

Teoria dei Numeri e Crittografia.

Note di Sandro Mattarei

Con alcune (poche) aggiunte di Andrea Caranti

Indice

Introduzione	4
Capitolo 1. Alcuni argomenti di Teoria dei Numeri elementare	5
1.1. Stime temporali per eseguire calcoli aritmetici	5
1.2. L'algoritmo di Euclide	15
1.3. Congruenze, il Teorema Cinese dei resti, la funzione di Eulero	18
1.4. Il teorema di Dirichlet sui primi in una progressione aritmetica	34
1.5. Cenni sulla distribuzione dei numeri primi	36
1.6. La moltiplicazione alla Montgomery	43
Capitolo 2. Campi finiti e resti quadratici	45
2.1. Campi finiti	45
2.2. Resti quadratici e reciprocità	50
2.3. Estrazione di radici quadrate modulari	61
Capitolo 3. Crittografia	71
3.1. La crittografia in generale	71
3.2. L'idea della crittografia a chiave pubblica	75
3.3. Il logaritmo discreto	85
Capitolo 4. Test di primalità	91
4.1. Premesse	91
4.2. Pseudoprimi	91
4.3. Pseudoprimi di Eulero	94
4.4. Pseudoprimi forti	97
Capitolo 5. Metodi di fattorizzazione	99
5.1. Il metodo $p - 1$ di Pollard	99
5.2. Fattorizzazione di Fermat	101
5.3. Il crivello quadratico (<i>quadratic sieve</i>)	102
Bibliografia	105

Introduzione

Queste note sono state scritte dapprima a mano da Sandro Mattarei per un corso tenuto a Trento nell'A.A. 1998/99, successivamente trascritte in \LaTeX da Claretta Carrara, e poi leggermente rivedute ed ampliate da Andrea Caranti. La fonte principale è [Kob94].

Il corso, tenuto normalmente da Sandro Mattarei, salvo una volta da Andrea Caranti ed una da Willem de Graaf, ha avuto luogo regolarmente ogni A.A.. Quasi ogni anno le note vengono leggermente ampliate e modificate durante il corso.¹ In realtà le note contengono più materiale di quanto venga effettivamente svolto nel corso, il quale varia di anno in anno.

In ogni momento (almeno fino alla fine dell'attuale A.A.) si potrà trovare la versione più aggiornata di queste note alla pagina WEB

<http://science.unitn.it/~mattarei/Didattica/Numeri/08-09/>

Le note sono basate soprattutto sul testo di Koblitz [Kob94], integrato da varie fonti. In particolare, sono stati utili [HW79], [IR90] e [NZ72] (Capitoli 1 e 2), [Coh93] (Capitoli 1, 2, 4 e 5), e [Rie94] (Capitoli 4 e 5).

¹Questa versione è aggiornata al 6 aprile 2009.

Alcuni argomenti di Teoria dei Numeri elementare

1.1. Stime temporali per eseguire calcoli aritmetici

1.1.1. Numeri in basi diverse. Fissato un intero positivo b , detto base, ogni numero intero non-negativo n si può scrivere in modo unico come

$$n = d_{k-1}b^{k-1} + d_{k-2}b^{k-2} + \dots + d_1b + d_0,$$

e sarà indicato con la notazione $(d_{k-1} \dots d_1 d_0)_b$ dove ciascuna d_i è una cifra in base b , cioè un simbolo per uno degli interi $0, 1, \dots, b-2, b-1$ (ad esempio $0, 1, \dots, b-1$ stessi se $b \leq 10$, oppure delle lettere). Se $d_{k-1} \neq 0$, diremo che n è un numero reale positivo si potrà scrivere come somma di una serie $\sum_{i < k} d_i b^i$ (dove i assume tutti i valori interi minori di k , quindi anche negativi) e indicare con $(d_{k-1} d_{k-2} \dots d_1 d_0, d_{-1} d_{-2} \dots)_b$. Anche qui la scrittura sarà unica, ad eccezione del caso in cui i d_i a partire da un certo d_j valgano $b-1$, in tal caso si potrà rimpiazzare ogni d_i con $i \geq j$ con 0 (e ometterlo nella scrittura) ed aumentare di uno il d_{j-1} . (Per una dimostrazione vedi piú avanti.)

1.1.2. Conversione da base b a base 10. Convieni impostare la conversione nel modo seguente (se n è intero):

$$n = (\dots ((d_{k-1}b + d_{k-2})b + d_{k-3})b + \dots + d_1)b + d_0.$$

Questo metodo (che conviene usare, analogamente, anche per calcolare il valore di un polinomio su un certo elemento) comporta soltanto $k-1$ moltiplicazioni per b e $k-1$ addizioni.

È anche facile da eseguire su un calcolatore tascabile (non troppo intelligente, cioè che esegua le operazioni nella sequenza in cui gli vengono richieste).

Ad esempio

$$(61405)_7 = ((6 \cdot 7 + 1) \cdot 7 + 4) \cdot 7 \cdot 7 + 5 = 14950.$$

Se n non è un intero una possibilità è proseguire lo stesso procedimento con le cifre (in base b) dopo la virgola, fino all'ultima d_{-h} (o fino alla precisione desiderata), ed infine dividere il risultato per b^h .

Notare che fino a prima di questo ultimo passaggio si lavora con numeri interi, mentre alla fine si può ottenere un decimale illimitato (supponendo di aumentare la precisione indefinitamente) anche partendo da un numero (razionale) con un numero finito di cifre in base b . Questo problema non si presenta naturalmente se $b = 2$, o piú in generale b ha solo 2 e 5 come fattori primi.

In realtà già conoscevate questo algoritmo per scrivere un intero da base 10 a base arbitraria. Piú in generale, permette di calcolare efficientemente il valore di

un polinomio per un dato valore numerico della indeterminata, ed è la ben nota *regola di Ruffini*. Nel caso dell'esempio abbiamo:

$$\begin{array}{r|rrrr} & 6 & 1 & 4 & 0 & 5 \\ & & 42 & 301 & 2135 & 14945 \\ \hline 7 & 6 & 43 & 305 & 2135 & 14950 \end{array}$$

1.1.3. Conversione da base 10 a base b . Se n è intero, si ottiene la sua ultima cifra in base b , cioè d_0 , come resto della divisione di n per b , poi d_1 come resto della divisione del quoziente precedente per b , e così via (se vogliamo, indefinitamente, comunque i resti saranno nulli da un certo punto in poi).

Ad esempio, per convertire $(14950)_{10}$ in base 7:

$$\begin{array}{r} \text{(resti parziali e) resto} \\ \text{quoziente} \end{array} \begin{array}{r} 14950 : 7 = \\ \hline 245 \\ 2135 : 7 = \\ \hline 30 \\ 305 : 7 = \\ \hline 24 \\ 43 : 7 = \\ \hline 1 \\ 6 : 7 = \\ \hline 6 \\ \hline 0 \end{array}$$

Il resto vero e proprio di ciascuna divisione (indicato in grassetto) è l'ultimo dei "resti parziali" (che qui consistono di singole cifre decimali, essendo $7 < 10$). Quindi $(14950)_{10} = (61405)_7$. Per eseguire questa procedura a mano su un calcolatore tascabile (che non fa la divisione con resto) possiamo semplicemente dividere n per b e confrontare la parte decimale con una tabellina preparata preventivamente, contenente $\frac{0}{b}, \frac{1}{b}, \frac{2}{b}, \dots, \frac{b-1}{b}$; quindi ripetere il procedimento rimpiazzando n con la parte intera del quoziente, e così via. Possiamo addirittura risparmiare la fatica di prendere la parte intera del quoziente ed usare il quoziente stesso. Nell'esempio precedente avremmo la tabella:

$$\begin{array}{l} 1/7=0,142 \dots \\ 2/7=0,285 \dots \\ 3/7=0,428 \dots \\ 4/7=0,571 \dots \\ 5/7=0,714 \dots \\ 6/7=0,857 \dots \end{array}$$

da confrontare con le parti decimali delle divisioni. Si ottiene

$$\begin{array}{l} 14950 \quad \cdot \\ 2135,714 \dots \\ 305,102 \dots \\ 43,586 \dots \\ 6,226 \dots \\ 0,889 \dots \end{array} \quad \text{e quindi} \quad \begin{array}{|c|c|} \hline 0,714 & 5 \\ \hline 0,102 & 0 \\ \hline 0,586 & 4 \\ \hline 0,226 & 1 \\ \hline 0,889 & 6 \\ \hline \end{array}$$

Notate che la tabella deve essere sufficientemente accurata, specie se vi sono sequenze di 0 o di 6 nella forma di n in base b . Non solo, anche le divisioni vanno eseguite con una precisione sufficiente, altrimenti si cade nell'errore che ora esemplifichiamo. Consideriamo il numero decimale 16806. Se facciamo le divisioni per 7 approximate a 3 cifre decimali, otteniamo (ricontrollare)

16806	.	0,857	6
2400,857	...	0,980	6
342,980	...	0,997	6
48,997	...	0,0	0
7,000	...	0,0	0
1	...	0,142	1
0,142	...		

Semberebbe quindi che $(16806)_{10} = (100666)_7$. Invece $(16806)_{10} = (66666)_7$. Il problema è che

$$\frac{6}{7} + \frac{6}{7^2} + \frac{6}{7^3} + \frac{6}{7^4} = 0.99958307\dots,$$

che si approssima a 1 a tre cifre decimali. (In altre parole, il 7.000 che appare nella penultima riga del nostro calcolo dovrebbe in realtà essere un po' meno di 7. Notiamo anche che, purtroppo, l'errore sul risultato finale non è nemmeno piccolo: in questo caso è di $(1000)_7 = 7^3$.)

Se n non è un intero possiamo trattare separatamente la parte intera e quella decimale. Se moltiplichiamo quest'ultima per b e prendiamo la parte intera del risultato otteniamo d_{-1} ; poi con la parte decimale del risultato ripetiamo il procedimento e otteniamo d_{-2} , e così via.

Ad esempio, convertendo π in base 2 avremo:

$$(3, 1415926\dots)_{10} = (11, 00100100001111110\dots)_2.$$

OSSERVAZIONI. (1) Per convertire da base b a base 10 e viceversa abbiamo usato due metodi diversi (l'uno l'inverso dell'altro: *moltiplicare* nel primo caso, *dividere* nel secondo). Per simmetria, ciascun metodo funzionerebbe anche nell'altra situazione, tuttavia sarebbe poco pratico, perché comporterebbe l'esecuzione di operazioni in base b anziché 10. Naturalmente se $b = 2$ e facciamo eseguire le conversioni ad un calcolatore che lavora in base 2, gli facciamo usare i due algoritmi scambiati.

(2) Le conversioni dal sistema binario (base 2) al sistema esadecimale (base 16, dove le cifre usate sono di solito $0, 1, \dots, 9, A, B, C, D, E, F$) e viceversa saranno naturalmente molto più semplici di quanto descritto. Ad esempio per convertire da binario a esadecimale basterà spezzare le cifre in blocchi di 4 a partire dalla virgola, ecc.

(3) Notate che il numero di cifre in base b di un intero non-negativo n è pari a

$$k = \lfloor \log_b n \rfloor + 1 = \left\lfloor \frac{\log n}{\log b} \right\rfloor + 1$$

(essendo $b^{k-1} \leq n < b^k$).

Il metodo che abbiamo visto per convertire da base 10 a base b si può anche rileggere come una dimostrazione che ogni numero reale positivo n si può scrivere in base b , cioè come somma di una serie $\sum_{i < k} d_i b^i$ con $0 \leq d_i < b$. Infatti una tale serie è sempre convergente (essendo a termini nonnegativi e maggiorata da $\sum_{i < k} (b-1)b^i = b^k$). D'altra parte, se i d_i sono le cifre in base b di n ricavate secondo la procedura descritta, la somma della serie è n . Per vederlo, basta maggiorare la differenza da n delle sue ridotte con potenze b^j , con $j \rightarrow -\infty$. Più precisamente, bisogna mostrare, per induzione "discendente" su j , che $\sum_{j \leq i < k} d_i b^i \leq n < \sum_{j \leq i < k} d_i b^i + b^j$, ovvero che $d_j b^j \leq n' < d_j b^j + b^j$, avendo posto $n' = n - \sum_{j < i < k} d_i b^i$. Notate che $0 \leq n' < b^{j+1}$ per ipotesi induttiva. Ora, il passo dell'algoritmo descritto che definisce d_j , diciamo per $j < 0$ (ma l'altro caso è analogo) prevede che si moltiplichino $n' b^{-j-1}$ per b e che se ne prenda la parte intera. In altre parole, $d_j := \lfloor n' b^{-j} \rfloor$, e quindi $d_j \leq n' b^{-j} < d_j + 1$, da cui $d_j b^j \leq n' < d_j b^j + b^j$, che è quanto volevamo mostrare.

Viceversa, ogni serie $\sum_{i < k} d_i b^i$ con $0 \leq d_i < b$ rappresenta un numero reale nonnegativo. Tuttavia, come già accennato, se solo un numero finito dei d_i è minore di $b-1$ (cioè se tutti i d_i valgono $b-1$ da un certo punto in poi), quel numero è anche rappresentato da un'altra serie, avente solo un numero finito dei d_i maggiori di zero. Vediamo ora che questa è l'unica eccezione: se $\sum_{i < k} d_i b^i = \sum_{i < k} e_i b^i$, sia j massimo tale che $d_j \neq e_j$, diciamo $d_j > e_j$, allora $0 = \sum_{i < k} d_i b^i - \sum_{i < k} e_i b^i = (d_j - e_j)b^j + \sum_{i < j} (d_i - e_i)b^i \geq (d_j - e_j)b^j - \sum_{i < j} (b-1)b^i = (d_j - e_j - 1)b^j \geq 0$ (essendo $(d_i - e_i) \geq -(b-1)$ per ogni i), e quindi $d_j = e_j + 1$, $d_i = 0$ ed $e_i = b-1$ per $i < j$. In particolare, otteniamo una corrispondenza biunivoca fra i reali positivi e le loro rappresentazioni $d_{k-1} \dots d_0, d_{-1} \dots$ in cui k è intero, $0 \leq d_i < b$ per ogni i , $d_{k-1} > 0$, e infiniti dei d_i sono minori di $b-1$. (Esercizio: mostrate che se le cifre del numero reale positivo n in base b sono ottenute con la procedura descritta, allora infinite di loro sono minori di $b-1$.)

1.1.4. Numeri periodici. L'espansione in base b di un numero razionale positivo c/d , con $(c, d) = 1$, è limitata (che potremmo anche interpretare come periodica con periodo 0) o periodica. Un modo semplicissimo per vederlo, se diamo per buona la correttezza (che comunque sarebbe un facile esercizio verificare) dell'algoritmo che abbiamo imparato alla scuola elementare per eseguire la divisione con resto, è notare che la successione delle cifre del quoziente si ripeterà non appena si ripetono i resti parziali (purché abbiamo superato la virgola), e ciò dovrà necessariamente avvenire essendo tali resti minori del divisore d . L'espansione sarà limitata se e solo se tutti i fattori primi di d dividono anche la base b (facile esercizio). Se è periodica concludiamo anche che il periodo è lungo al massimo $d-1$ (ma vedi la Proposizione seguente per un risultato più preciso). Inoltre potrà essere periodica pura (cioè la periodicità si verifica per tutte le cifre dopo la virgola) o mista (cioè c'è un antiperiodo); sarà pura se e solo se $(d, b) = 1$, come si deduce dalla Proposizione che stiamo per dimostrare.

Per *numero periodico puro in base b , di periodo f* (non necessariamente minimo) intendiamo un numero $0 \leq a < 1$ le cui cifre in base b a destra della virgola si ripetono in blocchi di f cifre, ed il blocco che si ripete non è della forma $(0, \dots, 0)$

oppure $(b-1, \dots, b-1)$. Notate che qui diciamo *periodo* non il blocco che si ripete, ma la sua lunghezza. (In contesti piú elementari il nostro *periodo* si chiamerebbe probabilmente *lunghezza del periodo*.) Stiamo inoltre escludendo esplicitamente da questa definizione i due casi in cui l'espansione è limitata, che abbiamo già osservato essere equivalenti. Ad esempio, $12,999 = 13,000 = 13$ in base dieci, e questo non lo consideriamo numero periodico puro.

PROPOSIZIONE 1.1. *Una frazione c/d con $c, d \in \mathbb{Z}$, $0 \leq c < d$, ridotta ai minimi termini, rappresenta un numero periodico puro in base b , di periodo f , se e solo se d divide $b^f - 1$.*

In particolare, il periodo minimo dell'espansione in base b di una frazione c/d con $(c, d) = 1$ e $(d, b) = 1$, è pari all'ordine moltiplicativo di b modulo d (cioè nell'anello $\mathbb{Z}/d\mathbb{Z}$).

DIMOSTRAZIONE. ¹ Supponiamo che l'espansione di c/d in base b sia periodica pura di periodo f . Poiché moltiplicare per b^f equivale a spostare la virgola a destra di f posti, avremo che la parte dell'espansione dopo la virgola di $b^f \cdot c/d$ è la stessa che per c/d , e quindi $b^f \cdot c/d = a + c/d$ per qualche intero a (che nel caso in cui $c < d$ è l'intero rappresentato in base b dalle prime f cifre b -arie dopo la virgola di c/d). Dunque $(b^f - 1) \cdot c/d$ è intero, pertanto $d \mid (b^f - 1)$, essendo $(c, d) = 1$. ²

Viceversa, se d divide $b^f - 1$, allora $(b^f - 1) \cdot c/d$ è intero, e quindi $b^f \cdot c/d$ ha la stessa espansione dopo la virgola che c/d . Poiché la prima espansione si ottiene dalla seconda semplicemente spostandone la virgola a destra di f posti, concludiamo che tale espansione è periodica, e che f è un periodo.

La seconda asserzione segue immediatamente dalla prima. \square

Poiché l'ordine moltiplicativo di un elemento invertibile di $\mathbb{Z}/d\mathbb{Z}$ divide $\varphi(d)$, dove φ è la funzione di Eulero che studieremo piú avanti, il periodo minimo dell'espansione in base b di una frazione c/d come nella Proposizione divide $\varphi(d)$. Questo rafforza di molto l'osservazione fatta in precedenza che il periodo minimo è minore di d .

¹Per completezza, consideriamo brevemente anche il caso in cui l'espansione b -aria è finita, anche se non esplicitamente richiesto, per semplicità solo nel caso delle frazioni del tipo $1/d$. Se tutti i divisori primi di d sono anche divisori di b , allora $1/d$ si scrive come un numero b -ario (sarebbe l'analogo di "decimale", "binario", "ternario", ecc., in base b) finito. Ad esempio, in base 10 si ha $1/2 = 0.5$, e $1/25 = 0.04$. In base 6 si ha $1/3 = (0.2)_6$. In generale, se tutti i divisori primi di d sono anche divisori di b , scegliamo il minimo n tale che d divida b^n . Dunque $b^n = a \cdot d$, e $1/d = a/b^n$, ove il termine di sinistra, scritto in base b , è un numero b -ario finito. La scelta minima di n garantisce che la n -esima cifra b -aria dopo la virgola sia l'ultima non nulla.

²Questo giustifica almeno in parte la "formula" (che certo conoscerete in base 10) per trovare la "frazione generatrice" di un numero periodico in base b : nel caso in cui la parte intera sia nulla (poi è facile farlo in generale), basta prendere il numero intero (che sopra abbiamo chiamato a) rappresentato (in base b) dalle prime f cifre b -arie dopo la virgola (quello che a scuola si chiama "il periodo", se f è il periodo minimo, ma chiaramente funziona anche se f è un qualsiasi periodo), e dividerlo per $b^f - 1$ (che a scuola suonerebbe come "un numero le cui cifre sono una sequenza di f volte la cifra $b - 1$ "). Non è difficile poi passare al caso generale in cui può comparire un "antiperiodo".

1.1.5. Operazioni bit. Vogliamo stimare il tempo necessario ad un calcolatore per svolgere delle operazioni. Ad esempio, l'addizione di due numeri (nel sistema binario):

$$\begin{array}{r} \text{riporti } 1 \ 1 \ 1 \ 1 \\ \hline 1 \ 0 \ 1 \ 1 \ 0 \ + \\ \quad 1 \ 1 \ 1 \ 0 \ = \\ \hline 1 \ 0 \ 0 \ 1 \ 0 \ 0 \end{array}$$

Diremo *operazione bit* (dove bit significa binary digit, cioè cifra binaria) l'operazione di sommare due cifre binarie, più un eventuale riporto costituito da un'altra cifra binaria, ed annotare il risultato fatto di due cifre binarie (di cui una è il nuovo riporto).

Quindi sommare due interi non-negativi di k cifre binarie (se uno ha meno cifre binarie dell'altro, gli aggiungiamo degli zeri davanti) richiede k operazioni bit.

Definiamo il tempo necessario per eseguire una certa operazione di tipo aritmetico come il numero di operazioni bit richieste. Questo perché il tempo necessario ad un computer per eseguire tale operazione è sostanzialmente proporzionale al numero di operazioni bit richieste (dove la costante di proporzionalità ovviamente dipende dal calcolatore usato, e non si tiene conto del tempo necessario per operazioni di tipo amministrativo, come accedere alla memoria, e in particolare, ricopiare dati da un posto all'altro, ecc.).

Vediamo un esempio di moltiplicazione:

$$\begin{array}{r} 1 \ 0 \ 0 \ 1 \ 1 \ \times \\ \quad 1 \ 1 \ 0 \ 1 \ = \\ \hline 1 \ 0 \ 0 \ 1 \ 1 \\ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \\ 1 \ 0 \ 0 \ 1 \ 1 \\ \hline 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \end{array}$$

In generale per moltiplicare un numero m di k cifre per un altro n di l cifre, con $k \geq l$, dobbiamo scrivere al più l copie di m (anche meno se certe cifre di n sono 0), ciascuna opportunamente traslata a sinistra (e del tempo necessario a queste traslazioni non si tiene conto), e addizionarle.

Quindi dobbiamo eseguire al più l (in realtà al più $l - 1$) addizioni di numeri di al più $k + l$ cifre (perché le somme parziali si allungano, ma la penultima ha al più $k + l - 1$ cifre, mentre il risultato finale avrà $k + l - 1$ o $k + l$ cifre).

In definitiva, complessivamente bastano $l(k + l)$ operazioni bit. Più semplicemente, poiché $k \geq l$, ne bastano $2kl$, una stima più semplice anche se meno precisa.

Con un po' più di attenzione si potrebbe ottenere la stima kl (anzi, $k(l - 1)$, se vogliamo essere ancor più precisi...) per il numero di operazioni bit, tenendo conto che ciascuna delle addizioni comporta in realtà soltanto k operazioni bit non banali.

Tuttavia questo guadagno di un fattore costante 2 non ha per noi una grande importanza, e la notazione che stiamo per introdurre lo trascurerà.³

³In realtà un calcolatore non esegue le operazioni in base due, ma in base una potenza di due (ad esempio $b = 2^{32}$ se lavora a 32-bit, ecc.), ma questo non fa una differenza sostanziale.

1.1.6. La notazione O -maiuscola.

DEFINIZIONE 1.2. Siano f, g due funzioni a valori reali definite sull'insieme delle r -uple di interi positivi (o eventualmente di interi positivi maggiori di una certa costante).

Supponiamo che esistano due costanti $B, C > 0$ tali che se per ogni j si ha $n_j > B$ allora

$$0 < f(n_1, n_2, \dots, n_r) < Cg(n_1, n_2, \dots, n_r).$$

In tal caso diremo che f è *limitata da* g e scriveremo $f = O(g)$. Una notazione equivalente è $f \ll g$, che ha vari vantaggi fra cui quello di poter essere rovesciata: $g \gg f$.

OSSERVAZIONI. (1) La notazione $f = O(g)$ può essere pensata come f è minore di un multiplo costante di g definitivamente. (Notate che la notazione O rappresenta una relazione, benché sia indicata come una funzione; in questo senso la notazione \ll è più chiara.) Osserviamo che da $f = O(g)$ e $g = O(h)$ segue che $f = O(h)$, ma non che $g = O(f)$ (dunque la relazione è transitiva (e, banalmente, anche riflessiva), ma non simmetrica). Useremo anche la notazione $f = O(g) = O(h)$ come abbreviazione di $f = O(g)$ e $g = O(h)$.

(2) Nel caso di una singola variabile n vale

$$f = O(g) \iff \limsup_{n \rightarrow \infty} \frac{f(n)}{g(n)} < +\infty$$

(e f, g sono definitivamente positive).

(3) Formalmente in $f = O(g)$ possiamo sempre rimpiazzare g con una funzione che cresce più velocemente di g . In pratica però vorremmo scegliere come g la stima migliore possibile per limitare f , compatibilmente con la preferenza per funzioni g di facile descrizione.

(4) Notate che se $f(n)$ è un polinomio di grado d e a coefficiente direttore positivo, allora $f(n) = O(n^d)$ (più in generale $f = O(g) \Rightarrow f + g = O(g)$, cioè nello stimare una somma possiamo trascurare gli addendi *che crescono meno di altri*).

(5) Esempi: $\log n = O(n^d)$ e $n^d = O(e^n)$, qualunque sia d reale positivo. Infatti la successione n^d/e^n tende a zero grazie al criterio del rapporto; per l'altra basta notare che $\log n < n$, e quindi ad esempio $(\log n)/n^2 \rightarrow 0$, da cui segue che anche $(\log n)/n^d = (2/d)(\log n)^{d/2}/(n^{d/2})^2 \rightarrow 0$. Altri

L'*operazione bit* va rimpiazzata da un'*operazione cifra*, diciamo, di due possibili tipi: l'addizione di due cifre (e un eventuale riporto precedente, che dà come output due cifre, la somma modulo b ed il riporto), e la moltiplicazione di due cifre (anch'essa con riporto). Queste procedure sono codificate nell'hardware del processore in modo tale che una moltiplicazione di due cifre impiega lo stesso tempo che la loro addizione.

Ad esempio, la moltiplicazione di due numeri di k ed l cifre in base $b > 2$ richiede non solo all'incirca kl addizioni cifra, ma anche all'incirca kl moltiplicazioni cifra. Il tempo impiegato è quindi un multiplo costante di kl (con la costante dipendente dalla macchina), esattamente come nel caso binario descritto nel testo.

esempi: $\log \log n = O(\log n)$; $n^2 \log n = O(n^{2+\varepsilon})$ (sottintendendo, per ogni $\varepsilon > 0$).

Ad esempio, se $f(n)$ indica il numero di cifre binarie di n , sarà $f(n) = O(\log n)$, e la stessa stima vale per qualsiasi base b fissata al posto di 2. Se però vogliamo tener conto anche della dipendenza da b , e $f(n, b)$ indica il numero di cifre di n in base b , sarà più preciso scrivere

$$f(n, b) = O\left(\frac{\log n}{\log b}\right)$$

(che comunque è $= O(\log n)$ essendo $\log b \geq \log 2$).

Per quanto abbiamo visto sull'addizione e moltiplicazione di due interi (in binario o in altra base fissata) avremo

$$\text{Tempo } ((k \text{ bit}) + (k \text{ bit})) = O(k),$$

$$\text{Tempo } ((k \text{ bit}) \cdot (l \text{ bit})) = O(kl).$$

(Notate che questa è una scrittura molto abbreviata. Ad esempio, la seconda scrittura sta per la lunga frase: “utilizzando l'algoritmo di moltiplicazione imparato alla scuola elementare (tradotto in binario, naturalmente) possiamo calcolare il prodotto di due numeri di k ed l bit in $O(kl)$ operazioni bit”. Dunque, l'argomento di “Tempo” è un algoritmo per calcolare quanto descritto brevemente fra parentesi; in particolare, usando un algoritmo diverso potremmo ottenere una stima diversa, vedi più avanti.) Possiamo anche riscrivere la seconda stima in termini dei due interi m, n che moltiplichiamo:

$$\text{Tempo } (m \cdot n) = O((\log m) \cdot (\log n)).$$

In particolare se i due interi hanno più o meno la stessa grandezza può convenire la stima

$$\text{Tempo } ((k \text{ bit}) \cdot (k \text{ bit})) = O(k^2).$$

Continueremo ad usare questa stima, ma esistono algoritmi molto più efficienti (che usano la *Trasformata di Fourier Veloce* (FFT)) per moltiplicare due numeri di k cifre in $O(k \log k \log(\log k))$ operazioni bit, il che è meglio di $O(k^{1+\varepsilon})$ per ogni $\varepsilon > 0$.

C'è un semplicissimo argomento che permette di passare da (circa) k^2 operazioni bit a $\frac{3}{4}k^2$. Per semplicità, supponiamo k pari, e scriviamo $k = 2h$. Scriviamo i due numeri da moltiplicare come $a \cdot 2^h + b$ e $c \cdot 2^h + d$, con a, b, c, d numeri a h bit. Allora

$$\begin{aligned} (a \cdot 2^h + b) \cdot (c \cdot 2^h + d) &= ac2^{2h} + (ad + bc)2^h + bd \\ &= ac2^{2h} + ((a + b) \cdot (c + d) - ac - bd)2^h + bd. \end{aligned}$$

Nell'ultima espressione abbiamo solo tre prodotti di numeri a $h = k/2$ (o $h+1$) bit (e alcune addizioni che danno solo contributi lineari alla complessità, oltre a degli shift di costo zero), dunque complessità dell'ordine di $3h^2 = (3/4)k^2$, rispetto al k^2 del metodo ordinario. Fin qui abbiamo solo migliorato il coefficiente da 1 a $3/4$, e quindi la complessità rimane $O(k^2)$, ma iterando riusciremo ad abbassare

l'esponente di k : per eseguire ciascuna delle 3 moltiplicazioni di numeri di $k/2$ bit si può applicare lo stesso artificio, e così via (finché è possibile). Se $2^s \leq k < 2^{s+1}$, possiamo applicare il metodo s volte, con una complessità dell'ordine di $(3/4)^s \cdot k^2 = 3^s$ operazioni bit, giungendo quindi alla stima $O(k^{\log_2 3})$, cioè circa $O(k^{1.58})$, per la moltiplicazione. Questo appena descritto è l'*algoritmo di Karatsuba* (1963).

OSSERVAZIONI. Se riuscissimo a ridurre la moltiplicazione $a \cdot 2^h + b$ per $c \cdot 2^h + d$ da quattro a sole due operazioni di interi di h bit, anziché tre come appena visto, la stima dopo le iterazioni si abbasserebbe addirittura a $O(k)$. Anche se non funziona proprio così, questo ci può dare una vaga idea di come la FFT riesca ad abbassare il tempo a $O(k^{1+\epsilon})$.

Per la sottrazione e la divisione con resto valgono le stesse stime che per l'addizione e la moltiplicazione.

Per la sottrazione $a - b$ fra due numeri a k bit, possiamo pensare che funzioni in questo modo. Sottrarre una cifra binaria da un'altra si fa sommando le due, e prendendo la cifra meno significativa del risultato, cioè buttando via il "riporto". Nel caso in cui si debba sottrarre la cifra 1 di b dalla cifra 0 di a , occorre "prendere in prestito" dalle cifre più significative. Questo si fa percorrendo le cifre di a verso sinistra, cambiando gli 0 in 1, finché non si trova un 1, e allora lo si cambia in 0, e ci si ferma. Ciò implica alcuni controlli gratuiti del tipo "Questa cifra è zero?", e complessivamente al più k cambiamenti di 0 in 1; anche se decidiamo di contare questi ultimi come operazioni bit, la complessità in notazione O non cambia.

Alternativamente, possiamo definire una variante di operazione bit che sottrae un bit da un altro (anziché addizionare), eventualmente *prendendo in prestito* un bit dalla colonna immediatamente precedente; naturalmente la sottrazione dovrà anche tener conto di un eventuale bit *preso in prestito* al passo prima. Sono necessarie al più k operazioni bit di questo tipo per eseguire una sottrazione fra numeri di k bit.

Per la divisione con resto di a per b , si tratta semplicemente di continuare a sottrarre da a dei traslati di b , come nell'esempio seguente (la divisione di 371 per 9).

$$\begin{array}{r}
 1\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 1 : 1\ 0\ 0\ 1 = 1\ 0\ 1\ 0\ 0\ 1 \\
 \underline{1\ 0\ 0\ 1} \\
 1\ 0\ 1\ 0 \\
 \underline{1\ 0\ 0\ 1} \\
 1\ 0\ 1\ 1 \\
 \underline{1\ 0\ 0\ 1} \\
 1\ 0
 \end{array}$$

Si può vedere che in generale dividere un numero di k bit per uno di l bit, con $l \leq k$ (altrimenti il quoziente è zero), richiede al più $k - l + 1$ sottrazioni, ciascuna delle quali coinvolge essenzialmente due numeri di l bit; complessivamente servono quindi $(k - l + 1)l$ operazioni bit, quindi in particolare al più $\leq kl$. Un modo meno preciso di pensare la stima, ma facile da ricordare, è che per la divisione vale la stessa stima della moltiplicazione, nel senso seguente: per dividere a per

b con resto ci vuole essenzialmente lo stesso tempo che per eseguire l'operazione inversa, cioè moltiplicare b per il quoziente q (ed aggiungere il resto). Dunque la stima per la divisione è $O((\log q) \cdot (\log b))$, ovvero $O((\log(a/b)) \cdot (\log b))$ (che in particolare è $O((\log a) \cdot (\log b))$, ma la forma piú precisa mostrerà la sua utilità quando stimeremo la complessità dell'algoritmo di Euclide).

ESEMPIO 1.3. Stimare il tempo necessario a convertire un intero n di k cifre binarie nella sua rappresentazione in base 10.

Per farlo dobbiamo dividere ripetutamente per $10 = (1010)_2$ i successivi quozienti. I resti che otteniamo, appartenenti all'insieme

$$\{0, 1, 10, 1, 100, 101, 110, 111, 1000, 1001\},$$

una volta tradotti in cifre decimali, saranno le cifre decimali di k a partire dalle unità. Il numero di divisioni è il numero di cifre decimali di n , cioè

$$\left\lceil \frac{\log n}{\log 10} \right\rceil + 1 = O(k),$$

ciascuna delle quali richiede $O(4k) = O(k)$ operazioni bit (poiché n ha k cifre e 10 ne ha 4). In definitiva :

$$\text{Tempo (convertire } n \text{ (da binario) a decimale)} = O(k^2) = O(\log^2 n).$$

ESEMPIO 1.4. Stimare il tempo necessario a convertire un intero n nella sua rappresentazione in base b . (Si sottintende sempre, in termini di operazioni bit; dunque possiamo pensare n originalmente assegnato in forma binaria, e sarà un calcolatore ad eseguire la conversione.)

Ce la caviamo con $O\left(\frac{\log n}{\log b}\right)$ divisioni, ciascuna delle quali richiede

$$O(\log n \cdot \log b)$$

operazioni bit, quindi in definitiva il tempo è $O(\log^2 n)$, indipendentemente da b (se b è grande ciascuna divisione impiega di piú, ma servono meno divisioni).

Si vede analogamente che il tempo per convertire n da base b a binario è ancora $O(\log^2 n)$, e quindi ci vuole sempre questo tempo per convertire da una base generica ad un'altra. Attenzione però, da binario ad esadecimale e viceversa si converte in un tempo nullo, come già osservato in precedenza.

ESEMPIO 1.5. Stimare il tempo necessario a calcolare $n!$.

Usiamo questo algoritmo: moltiplichiamo 2 per 3, quindi il risultato per 4, ecc., fino a moltiplicare per n . Perciò serviranno n moltiplicazioni ($n - 2$ in realtà), e il numero di cifre dei fattori in gioco è limitato superiormente dal numero di cifre del risultato finale $n!$. Per stimare quest'ultimo notiamo che il prodotto di n interi di al piú k bit ha al massimo nk bit, e quindi se n ha k bit, $n!$ ne ha $O(nk)$.

È facile mostrare che questa stima è la stima corretta, cioè non è esagerata. Infatti

$$(\text{numero di cifre binarie di } n!) \geq \sum_{i=2}^n \log_2 i \geq (n-1) \cdot \frac{1 + \log_2 n}{2}$$

(avendo $n \mapsto \log_2 n$ la concavità verso l'alto), ed è quindi stimato anche inferiormente da un multiplo costante di $n \log n$.⁴

Vediamo anche che non si perde nulla nella notazione O stimando il prodotto parziale generico con $n!$, in quanto già il numero di cifre binarie di $(n/2)!$, e quindi di tutti i prodotti parziali successivi, è stimato inferiormente da un multiplo costante di $n \log n$.

Dunque in ciascuna delle $n - 2$ moltiplicazioni stiamo moltiplicando un intero minore o uguale a $n!$, e quindi di $O(nk)$ bit, per uno minore o uguale a n , e quindi di $O(k)$ bit, impiegando perciò $O(nk^2)$ operazioni bit. Complessivamente perciò bastano $O(n^2 k^2)$ operazioni bit per calcolare $n!$, cioè:

$$\text{Tempo (calcolare } n!) = O(n^2 \log^2 n).$$

Per quanto visto tale stima non è migliorabile, in quanto il tempo per calcolare $n!$ (con l'algoritmo descritto) è stimato anche *inferiormente* da un multiplo costante di $n^2 \log^2 n$.

DEFINIZIONE 1.6. Un algoritmo per eseguire un calcolo a partire da interi n_1, n_2, \dots, n_i di k_1, k_2, \dots, k_i bit è detto *a tempo polinomiale* se il numero di operazioni bit richieste è $O(k_1^{d_1} k_2^{d_2} \dots k_i^{d_i})$ per opportuni interi d_1, \dots, d_i .

Gli algoritmi fin qui esaminati (sommare, moltiplicare, convertire in base b , ecc.) erano a tempo polinomiale ad eccezione del calcolo di $n!$ che non lo è. (Notate anche che la grandezza stessa di $n!$ non è polinomiale nel numero di bit di n .)

ESERCIZIO 1.7. Vale la formula (facilmente dimostrabile per induzione)

$$\sum_{j=1}^n j^3 = \left(\frac{n(n+1)}{2} \right)^2.$$

Stimare in funzione di n il tempo per calcolare il primo membro (che non è polinomiale) e il secondo membro (che è polinomiale).

1.2. L'algoritmo di Euclide

Lo ricordo. Serve per trovare il massimo comune divisore (a, b) di due interi positivi $a > b$. Iniziamo eseguendo la divisione con resto di a per b , ottenendo $a = q_1 b + r_1$ (sottinteso $0 \leq r_1 < b$); ripetiamo il procedimento con b e r_1 al posto di a e b , $b = q_2 r_1 + r_2$, e procediamo ricorsivamente dividendo r_i per r_{i+1} per $i \geq 1$, ottenendo $r_i = q_{i+2} r_{i+1} + r_{i+2}$, finché non otteniamo un resto $r_{j+1} = 0$. L'ultimo resto non nullo è (a, b) . Inoltre possiamo utilizzare ciascuna delle j uguaglianze corrispondenti alle divisioni con resti non nulli per esprimere il resto di quella divisione in termini dei resti delle due precedenti. Quindi, procedendo

⁴Il ragionamento visto è perfettamente sufficiente al nostro scopo, ma volendo il numero di cifre binarie di $n!$ si potrebbe stimare in modo ancor più preciso usando la disuguaglianza $\log(n!) \geq n \log n - n + 1$ (logaritmi naturali, cioè in base e , vedi l'*approssimazione di Stirling* accennata verso la fine di questo capitolo). Infatti, ne segue che $\log_2(n!) \geq n \log_2 n - (n-1) \log_2 e$, e quindi che il numero di bit di $n!$ è $\geq (1 - \varepsilon) n \log_2 n$ per n sufficientemente grande (cioè fissato a piacere $\varepsilon > 0$, esiste N tale che il numero di bit di $n!$ è $\geq (1 - \varepsilon) n \log_2 n$ per $n > N$).

ricorsivamente a ritroso, riusciamo ad esprimere (a, b) come $(a, b) = ua + vb$ per opportuni interi u, v . Quando si include nell'algoritmo il calcolo di u e v si parla di *algoritmo di Euclide esteso*.

PROPOSIZIONE 1.8. *L'algoritmo di Euclide fornisce il massimo comune divisore di $a > b$ in un numero finito di passi, in un tempo $O(\log^3 a)$.*

DIMOSTRAZIONE. Sappiamo già che l'algoritmo termina (poiché $r_{i+1} < r_i$) e che l'ultimo resto non nullo è il massimo comune divisore di a e b .

Dimostriamo la stima sul tempo mostrando che i resti decrescono rapidamente, e precisamente $r_{i+2} < \frac{1}{2}r_i$. Infatti pochè i resti sono strettamente decrescenti, tutti i quozienti sono strettamente positivi, quindi

$$r_i = q_{i+2}r_{i+1} + r_{i+2} \geq r_{i+1} + r_{i+2} > 2r_{i+2}.$$

Ne segue che ci sono al più $2 \cdot \lfloor \log_2 a \rfloor$ divisioni, che è $O(\log a)$.^{5 6}

Poiché ogni divisione coinvolge interi minori o uguali ad a , si esegue in $O(\log^2 a)$ operazioni bit da cui la tesi. \square

Vedremo nella prossima sottosezione che, stando attenti, la stima di $O(\log^3 a)$ operazioni bit per l'algoritmo di Euclide si può abbassare a $O(\log^2 a)$. Così è anche per l'algoritmo di Euclide esteso che trattiamo ora.

⁵Spesso in queste note è lasciato al lettore il compito di verificare in dettaglio certe affermazioni plausibili fatte. Per fugare eventuali dubbi che l'uso della notazione O sia una scienza inesatta, per una volta facciamo il conto preciso:

$$a > 2r_1 > 4r_3 > \cdots > 2^i r_{2i-1} > \cdots > 2^{\lfloor \frac{j+1}{2} \rfloor} r_{\hat{j}} \geq 1,$$

dove $\hat{j} = 2\lfloor \frac{j+1}{2} \rfloor - 1$, cioè $\hat{j} = j$ se j è dispari, $\hat{j} = j - 1$ se j è pari; in ogni caso, $j/2 \leq \lfloor \frac{j+1}{2} \rfloor \leq \lfloor \log_2 a \rfloor$.

⁶Il coefficiente 2 nella stima di $2 \cdot \lfloor \log_2 a \rfloor$ divisioni ottenuta nella dimostrazione, cioè nella velocità con cui decrescono i resti, non è ottimale. Infatti, la disuguaglianza $r_i > 2r_{i+2}$ trovata è solo la seconda di una serie di cui la prima è $r_i > r_{i+1}$, e di cui la terza e quarta sono le seguenti:

$$\begin{aligned} r_i &= q_{i+2}r_{i+1} + r_{i+2} = q_{i+2}(q_{i+3}r_{i+2} + r_{i+3}) + r_{i+2} \\ &= (q_{i+2}q_{i+3} + 1)r_{i+2} + q_{i+2}r_{i+3} \\ &\geq 2r_{i+2} + r_{i+3} > 3r_{i+3}, \end{aligned}$$

$$\begin{aligned} r_i &= (\cdots) = (q_{i+2}q_{i+3} + 1)(q_{i+4}r_{i+3} + r_{i+4}) + q_{i+2}r_{i+3} \\ &= ((q_{i+2}q_{i+3} + 1)q_{i+4} + q_{i+2})r_{i+3} + (q_{i+2}q_{i+3} + 1)r_{i+4} \\ &\geq 3r_{i+3} + 2r_{i+4} > 5r_{i+4}. \end{aligned}$$

Notate che, asintoticamente, ciascuna disuguaglianza è migliore della precedente: la seconda dice che, mediamente, ogni resto è più che $2^{1/2} \approx 1.414$ volte il successivo, e questo numero diventa $3^{1/3} \approx 1.442$ per la terza, e $5^{1/4} \approx 1.495$. Come potete intuire, le disuguaglianze successive sarebbero $r_i > 8r_{i+5}$, $r_i > 13r_{i+6}$, ecc., e hanno per coefficienti i numeri di Fibonacci. Poiché il rapporto fra numeri di Fibonacci consecutivi tende al *rapporto aureo* $(1 + \sqrt{5})/2 \approx 1.618$, ne deduciamo che il numero di passi dell'algoritmo di Euclide, quando è grande, ammette una stima per eccesso vicina a $(\log a)/(\log 1.618)$, che è un po' migliore della stima $(\log a)/(\log 1.414)$ trovata in precedenza.

PROPOSIZIONE 1.9. *Sia $d = (a, b)$ il massimo comune divisore di $a > b > 0$. Allora esistono interi u e v tali che $d = ua + vb$, e si possono trovare in $O(\log^3 a)$ operazioni bit.*

DIMOSTRAZIONE. Sostituendo ciascuna uguaglianza fornita dall'algoritmo nella precedente si riesce a scrivere l'ultimo resto non nullo $d = r_j$ come combinazione di r_{i-1} e r_i , per i a decrescere fino a $i = 0$ (dove $a = r_{-1}$, $b = r_0$).

Esplicitamente, iniziamo con $d = r_j = u_{j-1}r_{j-2} + v_{j-1}r_{j-1}$, avendo posto $(u_{j-1}, v_{j-1}) = (1, -q_j)$.

In generale, avendo scritto $d = u_i r_{i-1} + v_i r_i$ per un certo i , sostituiamo in essa $r_i = r_{i-2} - q_i r_{i-1}$, trovando

$$\begin{aligned} d &= u_i r_{i-1} + v_i (r_{i-2} - q_i r_{i-1}) \\ &= v_i r_{i-2} + (u_i - q_i v_i) r_{i-1}, \end{aligned}$$

Dunque la coppia $(u_{i-1}, v_{i-1}) = (v_i, u_i - q_i v_i)$ si ricava da (u_i, v_i) mediante una moltiplicazione e una sottrazione (anzi addizione, avendo u_i e v_i segno opposto, vedi sotto). Il procedimento si conclude con $d = u_0 a + v_0 b$, dunque ci sono $O(\log a)$ passi, ciascuno dei quali si fa in $O(\log^2 a)$ operazioni bit, in quanto tutti gli addendi e fattori coinvolti in valore assoluto non superano a .⁷ \square

OSSERVAZIONI. (1) Si vede facilmente che u_i e v_i hanno segno opposto per ogni i (infatti è vero per $i = j - 1$, e da $u_i v_i < 0$ segue $u_{i-1} v_{i-1} < 0$). Inoltre il segno di u_{i-1} è l'opposto di quello di u_i (e così per v_{i-1} e v_i). Quindi $u = u_0$ e $v = v_0$ hanno segno opposto (il che si vedeva anche direttamente) e il segno di u dipende dalla parità del numero di operazioni con cui si è concluso l'algoritmo.

(2) Si vede immediatamente che la coppia (u, v) è unica a meno di aggiungerle multipli di $(b/d, -a/d)$.

Ora supponete $d = 1$ per semplicità, e notate che la coppia (u_0, v_0) fornita dall'algoritmo di Euclide esteso soddisfa $|u_0| \leq b$ e $|v_0| \leq a$, come segue dalla dimostrazione (anzi, con il minore stretto, standoci piú attenti). Dunque, fra le possibili coppie (u, v) , quella fornita dall'algoritmo è una piuttosto speciale, essendo "quasi" l'unica (ce ne saranno un paio) che soddisfa tali condizioni (cioè essa è, in qualche modo, "la piú piccola"). (Il fatto che non sia proprio unica è un'imperfezione facilmente eliminabile, ad esempio imponendo che u abbia il segno giusto, oppure eseguendo le divisioni come spiego nell'osservazione successiva, e migliorando corrispondentemente (cioè dimezzando) le stime su u_0 e v_0 . La "quasi" unicità continua a valere anche se $d > 1$.)

(3) Ricordo che la divisione con resto si può anche modificare nel modo seguente: dati a e b interi positivi, esistono unici interi q ed r con $-b/2 < r \leq b/2$. Si può velocizzare l'algoritmo di Euclide eseguendo le divisioni

⁷Per verificare quest'ultima osservazione basta mostrare che $|u_i| \leq r_i$ e $|v_i| \leq r_{i-1}$ per induzione "discendente", partendo da $|u_{j-1}| = 1 \leq r_{j-1}$ e $|v_{j-1}| = q_j \leq r_{i-2}$. Concludiamo che per $i \geq 1$ valgono le disuguaglianze $|u_i| \leq r_1 < b$ e $|v_i| \leq b$, oltre naturalmente a $q_i \leq a$. Quindi per ciascuno dei passi bastano $O((\log a)(\log b))$ operazioni bit, che in particolare sono $O(\log^2 a)$.

in questo modo, essenzialmente dimezzando il numero di divisioni necessarie. (Non cambia nulla in termini della notazione O , ma può fare una differenza in pratica.)

1.2.1. Un miglioramento. Si può migliorare la stima della complessità dell'algoritmo di Euclide, e di quello esteso, abbassando l'esponente da 3 a 2 nel modo seguente.

La prima osservazione è che la complessità della divisione con resto di a per b (diciamo $a \geq b$) si può stimare come $O(\log b \log q)$, ove q è il quoziente. Infatti il costo è quello di tante sottrazioni a $\log b$ bit quanti sono i bit di q .

A questo punto si nota che nel calcolo del massimo comun divisore fra a e b si ha

$$\prod q_i < a,$$

ove i q_i sono i quozienti successivi. Infatti se $a = bq_1 + r_1$ si ha per cominciare

$$q_1 = \frac{a - r}{b} < a.$$

Se $b = r_1q_2 + r_2$, allora

$$q_1q_2 = \frac{a - r}{b} \cdot \frac{b - r_2}{r_1} = \frac{a - r}{r_1} \cdot \frac{b - r_2}{b} < a,$$

e così via.

Dunque la stima complessiva dell'algoritmo diventa

$$\log b \cdot (\log q_1 + \log q_2 + \cdots + \log q_n) = \log b \cdot \log \left(\prod q_i \right) < \log b \log a,$$

cioè $O((\log a)(\log b))$ (e quindi $O(\log^2 a)$).

In modo analogo si migliora a $O((\log a)(\log b))$ la stima dell'algoritmo di Euclide esteso. Dora in poi utilizzeremo ufficialmente queste stime più forti, piuttosto che quelle date dalle due Proposizioni precedenti.

1.3. Congruenze, il Teorema Cinese dei resti, la funzione di Eulero

Ricordo che un modo conveniente di lavorare con delle congruenze $a \equiv b \pmod{m}$ (che significa che m divide $(a - b)$), se m è fissato, è pensare una tale congruenza come l'uguaglianza delle classi resto modulo m di a e b (cioè $\bar{a} = \bar{b} \in \mathbb{Z}/m\mathbb{Z}$). È immediato in tal modo ricordare che le congruenze (rispetto ad uno stesso modulo) si possono addizionare, sottrarre e moltiplicare.

PROPOSIZIONE 1.10. *Gli elementi invertibili dell'anello $\mathbb{Z}/m\mathbb{Z}$ sono le classi resto rappresentate da interi primi con m (e ciò è indipendente dalla scelta del rappresentante). Inoltre se $(a, m) = 1$, un inverso b di a modulo m (cioè un rappresentante b della classe resto \bar{a}^{-1}) si può trovare in $O(\log^2 m)$ operazioni bit (assumendo $0 \leq a < m$).*

DIMOSTRAZIONE. Se \bar{a} è invertibile in $\mathbb{Z}/m\mathbb{Z}$, esiste $b \in \mathbb{Z}$ tale che $ab \equiv 1 \pmod{m}$, quindi $d = (a, m)$ dividendo sia a che m , divide anche 1. Viceversa se $(a, m) = 1$, esistono interi u, v tali che $ua + vb = 1$ e si possono trovare in $O(\log^2 m)$ operazioni bit (abbiamo assunto $0 \leq a < m$ perché ogni classe resto modulo m ha

un rappresentante non negativo e minore di m). Ora u è un inverso di a modulo m . \square

Se $(a, m) = 1$, con $a^{-1} \pmod{m}$ intenderemo un qualsiasi rappresentante della classe $\bar{a}^{-1} \in \mathbb{Z}/m\mathbb{Z}$.

COROLLARIO 1.11. *Sia assegnata da risolvere la congruenza $ax \equiv b \pmod{m}$, dove possiamo sempre assumere $0 \leq a, b < m$. Se $(a, m) = 1$ esiste una soluzione x_0 e si trova in $O(\log^2 m)$ operazioni bit. Ogni altra soluzione differisce da x_0 per un multiplo di m .*

Sia ora $d = (a, m)$ arbitrario. Allora esiste una soluzione se e solo se d divide b , ed in tal caso la congruenza è equivalente alla $a'x \equiv b' \pmod{m'}$ dove $a' = a/d$, $b' = b/d$, $m' = m/d$.

DIMOSTRAZIONE. La prima parte è chiara. Per la seconda, se esiste una soluzione x_0 , allora d , dividendo sia a che m , divide anche b . Viceversa se d divide b allora m divide $ax - b$ se e solo se m/d divide $(ax - b)/d$; a sua volta, la congruenza $a'x \equiv b' \pmod{m'}$ ha soluzioni grazie alla prima parte. \square

Ricordo che la funzione φ di Eulero è una funzione definita per n intero positivo nel modo seguente: $\varphi(n)$ è il numero di interi non negativi minori di n che sono coprimi con n . Equivalentemente, grazie alla Proposizione 1.10, $\varphi(n)$ è l'ordine del gruppo degli elementi invertibili dell'anello $\mathbb{Z}/n\mathbb{Z}$ (in quanto ogni classe resto ha un unico rappresentante non negativo minore di n). In particolare $\varphi(1) = 1$, e se n è una potenza di un primo p vale

$$\varphi(p^\alpha) = p^\alpha - p^{\alpha-1} = p^{\alpha-1} \cdot (p - 1) = p^\alpha \cdot \left(1 - \frac{1}{p}\right).$$

Questo perché l'anello $\mathbb{Z}/p^\alpha\mathbb{Z}$ ha un unico ideale massimale, precisamente $p\mathbb{Z}/p^\alpha\mathbb{Z}$, quindi gli elementi invertibili sono quelli che non gli appartengono.⁸

Oppure più semplicemente basta notare che gli interi primi con p^α sono quelli che non sono multipli di p .

1.3.1. Il Teorema Cinese dei Resti.

PROPOSIZIONE 1.12 (Teorema Cinese dei Resti). *Assegnato un sistema di congruenze $x \equiv a_i \pmod{m_i}$ per $i = 1, \dots, r$, con i moduli a due a due coprimi, cioè $(m_i, m_j) = 1$ per $i \neq j$, esiste sempre una soluzione, e due soluzioni differiscono per un multiplo di $m_1 m_2 \cdots m_r$.*

DIMOSTRAZIONE. Notate che l'omomorfismo di anelli

$$\begin{aligned} \mathbb{Z} &\rightarrow \mathbb{Z}/m_1\mathbb{Z} \times \cdots \times \mathbb{Z}/m_r\mathbb{Z} \\ n &\mapsto (n + m_1\mathbb{Z}, \dots, n + m_r\mathbb{Z}) \end{aligned}$$

⁸Si dice che $\mathbb{Z}/p^\alpha\mathbb{Z}$ è un *anello locale*. Ricordo che, in generale, gli ideali di $\mathbb{Z}/n\mathbb{Z}$ (ovvero i sottogruppi, dato che in questo caso ogni sottogruppo è automaticamente un ideale) sono esattamente i sottoinsiemi della forma $m\mathbb{Z}/n\mathbb{Z}$, con $m \mid n$, dunque sono in corrispondenza con i divisori m di n .

ha per nucleo l'insieme degli interi che sono multipli di tutti gli m_j simultaneamente, cioè multipli del prodotto $M = m_1 m_2 \dots m_r$. Ne segue che il monomorfismo

$$\mathbb{Z}/M\mathbb{Z} \rightarrow \mathbb{Z}/m_1\mathbb{Z} \times \dots \times \mathbb{Z}/m_r\mathbb{Z}$$

è anche suriettivo per il *lemma dei cassetti*. \square

DIMOSTRAZIONE COSTRUTTIVA. Poniamo $M = m_1 m_2 \dots m_r$, e $M_i = M/m_i$ per ogni i . Essendo $(M_i, m_i) = 1$ esiste per ogni i un intero N_i (e si può trovare con l'algoritmo di Euclide) tale che $M_i N_i \equiv 1 \pmod{m_i}$. D'altra parte è chiaro che $M_i N_i \equiv 0 \pmod{m_j}$ se $i \neq j$. Ne segue che

$$x_0 = \sum_{i=1}^r a_i M_i N_i$$

è una soluzione di $x \equiv a_i \pmod{m_i}$ per ogni i . \square

OSSERVAZIONI. (1) La strategia della seconda dimostrazione (che, al contrario della prima, è costruttiva, nel senso che fornisce un algoritmo per il calcolo di una soluzione) consiste nella ricerca, per ogni i , di un numero intero (che sarà $M_i N_i$) che nell'isomorfismo considerato nella prima dimostrazione corrisponda all'idempotente

$$(0 + m_1\mathbb{Z}, \dots, 0 + m_{i-1}\mathbb{Z}, 1 + m_i\mathbb{Z}, 0 + m_{i+1}\mathbb{Z}, \dots, 0 + m_r\mathbb{Z})$$

nel prodotto diretto di anelli. Il primo passo è trovare un intero (vale a dire M_i) che corrisponda ad un elemento del prodotto diretto fatto di zeri ad eccezione di un invertibile nell' i -esima componente.

- (2) In pratica, per risolvere un sistema di congruenze a mano, conviene operare ricorsivamente considerandone sempre due alla volta. Notate anche che per risolvere un sistema di due congruenze nel modo descritto dalla seconda dimostrazione basta un'esecuzione dell'algoritmo di Euclide anziché due. Infatti $M_1 = m_2$ e $M_2 = m_1$, ed una sola esecuzione produce N_1 ed N_2 tali che $m_2 N_1 + m_1 N_2 = 1$.
- (3) Spesso ci riferiremo col nome di teorema cinese dei resti all'isomorfismo costruito nella prima dimostrazione.

COROLLARIO 1.13. *La funzione φ è moltiplicativa, cioè*

$$\varphi(mn) = \varphi(m)\varphi(n) \quad \text{se } (m, n) = 1.$$

DIMOSTRAZIONE. Grazie all'isomorfismo $\mathbb{Z}/mn\mathbb{Z} \rightarrow \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$, il numero $\varphi(mn)$ di elementi invertibili del primo anello uguaglia quello del secondo, che è $\varphi(m)\varphi(n)$ per una proprietà generale del prodotto diretto di anelli. \square

Otteniamo quindi una formula per $\varphi(n)$ per qualunque n , basterà scrivere n come prodotto $n = p_1^{\alpha_1} \dots p_r^{\alpha_r}$ di potenze di primi distinti:

$$\varphi(n) = p_1^{\alpha_1} \left(1 - \frac{1}{p_1}\right) \cdot p_2^{\alpha_2} \left(1 - \frac{1}{p_2}\right) \dots p_r^{\alpha_r} \left(1 - \frac{1}{p_r}\right) = n \cdot \prod_{p|n} \left(1 - \frac{1}{p}\right).$$

Quindi possiamo calcolare $\varphi(n)$ con tale formula non appena conosciamo tutti i primi che dividono n , il che significa in pratica sapere scomporre n in fattori

primi. D'altra parte, la Proposizione seguente mostra come (almeno in un caso particolare), calcolare $\varphi(n)$ non può essere molto più semplice che fattorizzare n .

PROPOSIZIONE 1.14. *Sapendo che n è il prodotto di due primi distinti p e q , la conoscenza di p e q è sostanzialmente (nel senso computazionale) equivalente alla conoscenza di $\varphi(n)$. Più precisamente si calcola $\varphi(n)$ da p e q in $O(\log n)$ operazioni bit, e si calcolano p e q da $\varphi(n)$ in $O(\log^3 n)$ operazioni bit.*

DIMOSTRAZIONE. È banalmente vero se n è pari (e verificarlo non richiede alcun calcolo, lavorando in binario): $p = 2$, $q = n/2$, $\varphi(n) = n/2 - 1$.

Supponiamo n dispari. Allora $\varphi(n) = (p-1)(q-1) = n+1 - (p+q)$, che si calcola in $O(\log n)$ operazioni bit da p e q .

Viceversa, se conosciamo $\varphi(n)$ conosciamo oltre al prodotto $pq = n$ anche la somma $p+q = n+1 - \varphi(n)$, diciamola $2b$ visto che è pari. Allora p e q sono le due radici dell'equazione di secondo grado $x^2 - 2bx + n = 0$, date da $b \pm \sqrt{b^2 - n}$. Delle varie operazioni coinvolte quella più lunga è la radice quadrata, che richiede $O(\log^3 n)$ operazioni bit.

Infatti, per calcolare $\lfloor \sqrt{m} \rfloor$ dove m è un intero non negativo di k bit, si può partire da $2^{\lfloor (k-1)/2 \rfloor}$ come prima approssimazione, quindi determinare ciascun bit partendo dai più significativi ponendo ciascuna cifra pari a 1, elevando al quadrato e confrontando il risultato con n . Ciò si fa in $O(\log^3 n)$ operazioni bit. \square

È vero che dalla Proposizione 1.14 appare leggermente più semplice ricavare $\varphi(n)$ dalla fattorizzazione di n che viceversa, tuttavia questa piccola differenza (di esponenti diversi in tempi polinomiali) scompare al confronto con la complessità del problema di fattorizzare n , che è non-polinomiale con tutti i metodi noti (anche se l'impossibilità di farlo in tempo polinomiale non è mai stata dimostrata). Vediamo nell'esempio seguente la complessità del metodo più ingenuo di fattorizzazione.

ESEMPIO 1.15. Per scoprire almeno un fattore proprio (dopodiché si può ripetere il procedimento sui fattori già trovati) di un intero n basta provare a dividerlo per tutti gli interi positivi che non superano \sqrt{n} (e in caso di fallimento si può concludere che n è primo): ciò comporta $O(n^{1/2} \log^2 n)$ operazioni bit. (Sarebbero $O(n \log^2 n)$ se dovessi dividere per tutti gli interi minori di n .)

Se ci limitiamo a dividere per i primi che non superano \sqrt{n} (supponendo di averne una lista a disposizione), la stima si abbassa a $O(n^{1/2} \log n)$, in quanto il numero di divisioni da eseguire si riduce all'incirca di un fattore $\log(\sqrt{n})$. (Stiamo usando il Teorema dei numeri primi, di cui si parla più avanti, secondo il quale il numero di primi minori di m è circa $m/\log m$, in un senso da precisare.)

Anche se il metodo di fattorizzazione appena descritto è poco efficiente, esso viene normalmente usato come primo attacco alla fattorizzazione di un intero, prima di passare all'uso di metodi più efficienti, ma anche più complessi.

In pratica, non volendo memorizzare una lista di primi si può limitarsi, ad esempio, a dividere per 2, 3, 5 e poi per tutti gli interi primi con $2 \cdot 3 \cdot 5 = 30$, cioè per tutti gli interi congrui a $\pm 1, \pm 7, \pm 11, \pm 13$ modulo 30, senza preoccuparsi che parte di questi non siano primi e quindi diano verifiche superflue. Infatti, mentre

escludere le divisioni (superflue) per i numeri pari maggiori di 2 fa diminuire il tempo necessario di un fattore due, cioè lo moltiplica per $1/2$, escludere anche i multipli di 3 moltiplica il tempo per un ulteriore fattore $1 - \frac{1}{3} = \frac{2}{3}$, ed escludere anche i multipli di 5 moltiplica il tempo per un ulteriore fattore $1 - \frac{1}{5} = \frac{4}{5}$. Ad ogni passo il guadagno di tempo diventerebbe sempre piú marginale rispetto alla complicazione dell'algoritmo, ed a un certo tempo non conviene piú. Per capirci, mentre escludere le divisioni per i multipli di 2, 3, 5 (ad eccezione degli stessi 2, 3, 5, naturalmente) moltiplica il tempo necessario per un fattore $\frac{\varphi(2 \cdot 3 \cdot 5)}{2 \cdot 3 \cdot 5} = (1 - \frac{1}{2})(1 - \frac{1}{3})(1 - \frac{1}{5}) = \frac{4}{15}$, cioè circa 0,27, rispetto a dividere per tutti gli interi fino ad un certo limite, escludere le divisioni per i multipli di 2, 3, 5, 7 (ad eccezione di 2, 3, 5, 7) moltiplica il tempo necessario per un fattore $\frac{\varphi(2 \cdot 3 \cdot 5 \cdot 7)}{2 \cdot 3 \cdot 5 \cdot 7} = \frac{4}{15}(1 - \frac{1}{7}) = \frac{24}{105}$, cioè circa 0,23, non molto piú basso.

Quale sia la variante che scegliamo, le stime del metodo descritto nell'Esempio sono comunque non polinomiali (in $\log n$). Piú precisamente, si tratta di stime esponenziali: infatti $n^{1/2} \log^2 n = e^{\frac{1}{2} \log n + 2 \log \log n} = O(e^{(\frac{1}{2} + \varepsilon) \log n})$. Per un confronto, la stima per uno dei piú potenti metodi di fattorizzazione noti, il *quadratic sieve* (o *crivello quadratico*, si veda il Capitolo 5), è $O(e^{(1 + \varepsilon) \sqrt{\log n \log \log n}})$, dunque sempre non polinomiale, benché migliore di una stima esponenziale (si dice infatti *subesponenziale*).

1.3.2. La struttura moltiplicativa dell'anello $\mathbb{Z}/m\mathbb{Z}$.

PROPOSIZIONE 1.16 (Teorema di Eulero-Fermat). *Se $(a, m) = 1$, allora vale $a^{\varphi(m)} \equiv 1 \pmod{m}$.*

DIMOSTRAZIONE. Se $(a, m) = 1$, l'intero a rappresenta un elemento invertibile del gruppo $U(\mathbb{Z}/m\mathbb{Z})$ (indicato anche con $(\mathbb{Z}/m\mathbb{Z})^*$, da non confondere però con l'insieme degli elementi non nulli di $\mathbb{Z}/m\mathbb{Z}$, che in generale non è un gruppo) degli elementi invertibili dell'anello $\mathbb{Z}/m\mathbb{Z}$. Poiché tale gruppo ha ordine $\varphi(m)$ segue la tesi. \square

COROLLARIO 1.17 (Piccolo Teorema di Fermat). *Per ogni intero a e primo p vale $a^p \equiv a \pmod{p}$.*

DIMOSTRAZIONE. Se $p \mid a$ è immediato, se $p \nmid a$ segue dal Teorema di Eulero-Fermat. Si vede ancor meglio scrivendo la conclusione nella forma $a(a^{p-1} - 1) \equiv 0 \pmod{p}$. \square

Esistono altre dimostrazioni del Piccolo Teorema di Fermat. Ad esempio, chi non conosca i gruppi (e quindi il Teorema di Lagrange) lo può dimostrare per induzione (per a positivo, ma poi segue subito per a intero) basandosi sul fatto che $(a+1)^p \equiv a^p + 1 \pmod{p}$, che dipende solo dal fatto che p divide $\binom{p}{i} = \frac{p!}{i!(p-i)!}$ per $0 < i < p$. Un'altra dimostrazione del Piccolo Teorema di Fermat si ottiene contando le collane diverse di n perline che si possono formare usando perline di a colori diversi (per $a > 0$): sui vertici di un p -agone regolare si possono sistemare p perline in a^p modi diversi; queste disposizioni però si possono raggruppare in insiemi di p o $2p$ che corrispondono alla stessa collana a meno di rotazioni e

riflessioni (qui si usa che p è primo), ad eccezione delle a collane composte da un solo colore; ne segue che $a^p - a$ è multiplo di p .

È chiaro dalla dimostrazione data che l'enunciato del Teorema di Eulero-Fermat continua a valere rimpiazzando $\varphi(m)$ con qualunque intero f che sia multiplo degli ordini di tutti gli elementi di $U(\mathbb{Z}/m\mathbb{Z})$. (Detto in questa forma, il Teorema di Eulero-Fermat diviene quasi una tautologia.) In generale, si dice *esponente* di un gruppo finito G il più piccolo intero positivo f tale che $g^f = 1$ per ogni $g \in G$; equivalentemente, l'esponente di G è il minimo comune multiplo degli ordini di tutti gli elementi di G (quindi, in particolare, divide l'ordine di G). Se poi G è abeliano, il caso che ci interessa, esso ha sempre un elemento di ordine uguale al suo esponente (come segue facilmente dal fatto che $|gh| = |g| \cdot |h|$ se g ed h commutano e $(|g|, |h|) = 1$, come mostriamo nella prossima sottosezione), e quindi in questo caso l'esponente di G è anche il *massimo* degli ordini dei suoi elementi. Dunque il più piccolo esponente che messo al posto di $\varphi(m)$ nel Teorema di Eulero-Fermat ne conserva la validità è l'esponente e del gruppo $U(\mathbb{Z}/m\mathbb{Z})$. Il calcolo preciso di tale esponente si deduce dalla prossima Proposizione, che descrive, modulo il Teorema cinese dei resti, la struttura del gruppo $U(\mathbb{Z}/m\mathbb{Z})$. (Questa ci sarà utile varie volte nel seguito.)

Il primo passo per analizzare la struttura del gruppo $U(\mathbb{Z}/m\mathbb{Z})$ (e, come conseguenza, per abbassare l'esponente nel Teorema di Eulero-Fermat) si fa applicando il Teorema cinese dei resti. Scrivendo $m = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_r^{\alpha_r}$ con $p_1 < p_2 < \dots < p_r$, grazie al Teorema cinese dei resti (o alla sua dimostrazione) avremo $U(\mathbb{Z}/m\mathbb{Z}) \cong U(\mathbb{Z}/p_1^{\alpha_1}\mathbb{Z}) \times \dots \times U(\mathbb{Z}/p_r^{\alpha_r}\mathbb{Z})$. Ne deduciamo che l'esponente e di $U(\mathbb{Z}/m\mathbb{Z})$ divide il minimo comune multiplo di

$$\varphi(p_1^{\alpha_1}) = (p_1 - 1)p_1^{\alpha_1 - 1}, \dots, \varphi(p_r^{\alpha_r}) = (p_r - 1)p_r^{\alpha_r - 1}.$$

Quindi, se diciamo f tale minimo comune multiplo, avremo $a^f \equiv 1 \pmod{m}$ per ogni $a \in \mathbb{Z}$ con $(a, m) = 1$.

Possiamo anche dimostrare quest'ultimo fatto direttamente, senza usare il Teorema cinese dei resti, ma solo il suo corollario, la moltiplicatività di φ : essendo $a^{\varphi(p_i^{\alpha_i})} \equiv 1 \pmod{p_i^{\alpha_i}}$ per ogni i , e poiché f è multiplo di ciascun $\varphi(p_i^{\alpha_i})$, avremo $a^f \equiv 1 \pmod{p_i^{\alpha_i}}$ per ogni i , da cui segue la conclusione.

ESEMPIO 1.18. È facile calcolare $2^{1000000} \pmod{77}$ conoscendo la scomposizione di 77 in fattori primi, $77 = 7 \cdot 11$. Infatti questo ci permette di calcolare $\varphi(77) = (7 - 1)(11 - 1) = 60$, e quindi $2^{60} \equiv 1 \pmod{77}$. La discussione precedente mostra anzi che $2^{30} \equiv 1 \pmod{77}$, in quanto 30 è il minimo comune multiplo di $\varphi(7) = 6$ e $\varphi(11) = 10$ (e quindi l'esponente di $U(\mathbb{Z}/77\mathbb{Z})$ divide 30). Essendo $1000000 = 30 \cdot 33333 + 10$, avremo

$$2^{1000000} \equiv 2^{10} \equiv 23 \pmod{77}.$$

Un altro modo sarebbe stato calcolare separatamente $2^{1000000} \pmod{7}$ e $2^{1000000} \pmod{11}$ (con lo stesso metodo, trovando 2 e 1 rispettivamente) e poi usare il Teorema cinese dei resti per trovare l'unico intero non negativo minore di 77 che è congruo a 2 modulo 7 e congruo a 1 modulo 11. Anche in questo caso serviva comunque la scomposizione di 77.

Vedremo presto che se $p_1 > 2$ (cioè se m è dispari) l'esponente di $U(\mathbb{Z}/m\mathbb{Z})$ non solo *divide*, ma *eguaglia*, il minimo comune multiplo degli ordini $\varphi(p_1^{\alpha_1}), \dots, \varphi(p_r^{\alpha_r})$ dei vari gruppi $U(\mathbb{Z}/p_i^{\alpha_i}\mathbb{Z})$ coinvolti. Il punto cruciale è che questi gruppi sono ciclici, come vedremo nella Proposizione 1.20. Per dimostrarla abbiamo bisogno del seguente Lemma.

LEMMA 1.19. *Se p è dispari e $p \nmid a$, la classe di $1 + ap$ ha ordine $p^{\alpha-1}$ in $U(\mathbb{Z}/p^\alpha\mathbb{Z})$.*

Se $2 \nmid a$, la classe di $1 + 4a$ ha ordine $2^{\alpha-2}$ in $U(\mathbb{Z}/2^\alpha\mathbb{Z})$.

DIMOSTRAZIONE. Entrambe le conclusioni seguono dalla congruenza

$$(*) \quad (1 + ap)^{p^{\beta-2}} \equiv 1 + ap^{\beta-1} \pmod{p^\beta},$$

che vale per $\beta \geq 2$, e per a intero arbitrario se $p > 2$, ma solo per a intero pari se $p = 2$. Ad esempio, nel caso in cui p è dispari, la congruenza (*) con $\beta = \alpha + 1$ implica che $(1 + ap)^{p^{\alpha-1}} \equiv 1 \pmod{p^\alpha}$, e quindi la classe di $1 + ap$ ha ordine un divisore di $p^{\alpha-1}$ in $U(\mathbb{Z}/p^\alpha\mathbb{Z})$; la congruenza (*) con $\beta = \alpha$ poi mostra che tale ordine non divide $p^{\alpha-2}$, e quindi vale $p^{\alpha-1}$.

Ora dimostriamo la congruenza (*) per induzione su β . Infatti la congruenza si riduce ad un'identità per $\beta = 2$, mentre se vale per un certo $\beta \geq 2$, cioè se $(1 + ap)^{p^{\beta-2}} = 1 + ap^{\beta-1} + kp^\beta = 1 + p^{\beta-1}(a + kp)$ per un opportuno intero k , allora

$$\begin{aligned} (1 + ap)^{p^{\beta-1}} &= (1 + p^{\beta-1}(a + kp))^p \\ &= 1 + \binom{p}{1} p^{\beta-1}(a + kp) + \dots + p^{p(\beta-1)}(a + kp)^p \\ &\equiv 1 + ap^\beta \pmod{p^{\beta+1}}. \end{aligned}$$

Nell'ultimo passaggio abbiamo usato il fatto che $\binom{p}{i} p^{i(\beta-1)}$ è multiplo di $p^{\beta+1}$ per $1 < i < p$ (e $\beta \geq 2$), mentre $p^{p(\beta-1)}$ è multiplo di $p^{\beta+1}$ se e solo se $(p-1)\beta \geq p+1$, che quindi vale, ad eccezione del caso $(p, \beta) = (2, 2)$. In questo caso eccezionale il passo induttivo è $(1 + 2a)^2 \equiv 1 + 4a \pmod{8}$, che è vero se e solo se a è pari. \square

In particolare, grazie al Lemma 1.19, per p primo dispari la classe di $1 + p$ è un elemento di ordine $p^{\alpha-1}$ in $U(\mathbb{Z}/p^\alpha\mathbb{Z})$, e per p primo arbitrario la classe di $1 + p^2$ è un elemento di ordine $p^{\alpha-2}$ in $U(\mathbb{Z}/p^\alpha\mathbb{Z})$.

Dalla congruenza (*) mostrata nelle precedente dimostrazione possiamo in realtà dedurre il fatto seguente, che estende l'enunciato del Lemma 1.19. Se p è un primo dispari ed a è un intero arbitrario, allora la classe di $1 + ap$ ha ordine $p^{\alpha-\gamma-1}$ in $U(\mathbb{Z}/p^\alpha\mathbb{Z})$, dove p^γ è la massima potenza di p che divide a . L'enunciato analogo per $p = 2$ vale solo per a pari (cioè per $\gamma > 0$).

Notate che questo ragionamento produce $p^{\alpha-1}$ elementi di $U(\mathbb{Z}/p^\alpha\mathbb{Z})$ di ordine una potenza di p (quelli rappresentati da interi congrui a 1 modulo p). Poichè $U(\mathbb{Z}/p^\alpha\mathbb{Z})$ è un gruppo abeliano, di ordine $\varphi(p^\alpha) = (p-1)p^{\alpha-1}$, quelli considerati devono essere *tutti* gli elementi di ordine una potenza di p . (Questo segue dal fatto generale che un gruppo abeliano G di ordine mn con $(m, n) = 1$ è prodotto diretto interno di un sottogruppo di ordine m ed uno di ordine n .) Nel caso $p = 2$,

tutti gli elementi di $U(\mathbb{Z}/2^\alpha\mathbb{Z})$ hanno ordine una potenza di 2; quelli considerati nel paragrafo precedente, cioè quelli rappresentati da interi congrui a 1 modulo 4, sono la metà del totale, e l'altra metà sono quelli rappresentati da interi congrui a -1 modulo 4. (Su questo punto si veda la dimostrazione della proposizione seguente.)

OSSERVAZIONI. Un punto di vista piú sofisticato sul contenuto del Lemma 1.19 (da cui, volendo, si ottiene una dimostrazione alternativa) è il seguente. Si può mostrare che per $p > 2$ la mappa

$$\log : 1 + ap \mapsto \log(1 + ap) = \sum_{j=1}^{\infty} (-1)^{j-1} \frac{a^j p^j}{j}$$

è ben definita per $1 + ap \in (1 + p\mathbb{Z})/p^\alpha\mathbb{Z}$, e manda il sottogruppo moltiplicativo $(1 + p\mathbb{Z})/p^\alpha\mathbb{Z}$ di $U(\mathbb{Z}/p^\alpha\mathbb{Z})$ (che l'immagine di $1 + p\mathbb{Z}$ in $\mathbb{Z}/p^\alpha\mathbb{Z}$) biiettivamente sul sottogruppo additivo $p\mathbb{Z}/p^\alpha\mathbb{Z}$ di $\mathbb{Z}/p^\alpha\mathbb{Z}$. (In realtà solo un numero finito dei termini della serie infinita dà un contributo non nullo.) Per una proprietà formale del logaritmo, tale mappa è anzi un isomorfismo di gruppi. (Inoltre, la sua inversa è la mappa $\exp : bp \mapsto \exp(bp) = \sum_{j=0}^{\infty} b^j p^j / j!$, per $bp \in p\mathbb{Z}/p^\alpha\mathbb{Z}$.) In particolare, il gruppo moltiplicativo $(1 + p\mathbb{Z})/p^\alpha\mathbb{Z}$ è ciclico, essendolo il gruppo additivo $p\mathbb{Z}/p^\alpha\mathbb{Z}$.

Per $p = 2$ invece la mappa \log produce solo un isomorfismo di gruppi di $(1 + 4\mathbb{Z})/2^\alpha\mathbb{Z}$ su $4\mathbb{Z}/2^\alpha\mathbb{Z}$. In effetti essa è definita su $(1 + 2\mathbb{Z})/2^\alpha\mathbb{Z}$, ma manda questo su $4\mathbb{Z}/2^\alpha\mathbb{Z}$ con nucleo $(\pm 1 + 2\mathbb{Z})/2^\alpha\mathbb{Z}$. (Che -1 appartenga al nucleo corrisponde al seguente fatto, che non è né ovvio, né facile da dimostrare direttamente: per ogni potenza 2^t di 2, esiste s_0 tale che il numeratore del numero razionale $\sum_{j=1}^s 2^j/j$ sia divisibile per 2^t per ogni $s \geq s_0$.) D'altra parte, si vede facilmente che la mappa \exp è definita su $4\mathbb{Z}/2^\alpha\mathbb{Z}$, ma non lo è su $2\mathbb{Z}/2^\alpha\mathbb{Z}$: la serie $\sum_{j=0}^{\infty} 2^j/j!$ ha infiniti termini divisibili per 2 ma non per 4, e precisamente quelli dove j ha la forma $j = 2^u$.

Per chi volesse approfondire, questi argomenti si studiano nell'Analisi p -adica.

PROPOSIZIONE 1.20. *Il gruppo $U(\mathbb{Z}/p^\alpha\mathbb{Z})$ è ciclico (e quindi ha esponente uguale al suo ordine) se p è un primo dispari. Invece $U(\mathbb{Z}/2^\alpha\mathbb{Z})$ non è ciclico per $\alpha \geq 3$, bensì è il prodotto diretto di un gruppo ciclico di ordine 2 per un altro ciclico di ordine $2^{\alpha-2}$, generati ad esempio da $-\bar{1}$ e $\bar{5}$.*

DIMOSTRAZIONE. Sia p un primo dispari. Anticipiamo dal prossimo Capitolo il fatto che il gruppo $U(\mathbb{Z}/p\mathbb{Z})$ è ciclico, essendo il gruppo moltiplicativo del campo con p elementi \mathbb{F}_p . Ci sono vari modi per dedurre da questo fatto che anche $U(\mathbb{Z}/p^\alpha\mathbb{Z})$ è ciclico.

Una possibilità è sfruttare l'omomorfismo di gruppi $\psi : U(\mathbb{Z}/p^\alpha\mathbb{Z}) \rightarrow U(\mathbb{Z}/p\mathbb{Z})$, anch'esso suriettivo, ottenuto per restrizione dall'omomorfismo di anelli suriettivo $\mathbb{Z}/p^\alpha\mathbb{Z} \rightarrow \mathbb{Z}/p\mathbb{Z}$ tale che $a + p^\alpha\mathbb{Z} \mapsto a + p\mathbb{Z}$. Se $g + p\mathbb{Z}$ è un generatore di $U(\mathbb{Z}/p\mathbb{Z})$, allora $g + p^\alpha\mathbb{Z}$ (che appartiene alla sua controimmagine $\psi^{-1}(g + p\mathbb{Z})$) ha ordine un multiplo di $p - 1$ (perché, in generale, se $\psi : G \rightarrow H$ è un omomorfismo allora l'ordine di $\psi(g)$ divide l'ordine di g). Possiamo quindi ricavarne un elemento di ordine (esattamente) $p - 1$ in $U(\mathbb{Z}/p^\alpha\mathbb{Z})$, ad esempio $g^{p^{\alpha-1}} + p^\alpha\mathbb{Z}$ (confrontate

con la dimostrazione costruttiva che \mathbb{F}_q^* è ciclico, all'inizio del secondo Capitolo). Avremmo potuto fare questo ragionamento direttamente con le congruenze, senza nominare l'omomorfismo ψ , ma è meglio che ci abituiamo a lavorare con gruppi e omomorfismi, per prepararci a quando tali strumenti non si potranno evitare tanto facilmente.

Grazie al Lemma 1.19, in $U(\mathbb{Z}/p^\alpha\mathbb{Z})$ c'è anche un elemento di ordine $p^{\alpha-1}$, ad esempio la classe di $1+p$. Moltiplicando quest'ultimo per l'elemento di ordine $p-1$ trovato in precedenza otteniamo (poiché i due ordini sono coprimi, si veda la prossima Sottosezione) un elemento di ordine $(p-1)p^{\alpha-1} = |U(\mathbb{Z}/p^\alpha\mathbb{Z})|$, dimostrando quindi che $U(\mathbb{Z}/p^\alpha\mathbb{Z})$ è ciclico per p dispari.

Occupiamoci ora del caso $p=2$. Essendo $U(\mathbb{Z}/2\mathbb{Z})$ e $U(\mathbb{Z}/4\mathbb{Z})$ ovviamente ciclici, consideriamo dunque $U(\mathbb{Z}/2^\alpha\mathbb{Z})$ con $\alpha \geq 3$. È facile mostrare che esso non può essere ciclico, ad esempio perché ha esattamente tre elementi di ordine 2, precisamente $\overline{-1}$ e $\overline{2^{\alpha-1} \pm 1}$. (Verificalo, ad esempio risolvendo $x^2 - 1 = (x-1)(x+1) \equiv 0 \pmod{2^\alpha}$.)

Grazie al Lemma 1.19, la classe di 5 ha ordine $2^{\alpha-2}$ in $U(\mathbb{Z}/2^\alpha\mathbb{Z})$. Ora è facile mostrare che $U(\mathbb{Z}/2^\alpha\mathbb{Z}) = \langle \overline{-1} \rangle \times \langle \overline{5} \rangle$, ad esempio nel modo seguente. Le potenze $\overline{5^0}, \overline{5^1}, \overline{5^2}, \dots, \overline{5^{2^{\alpha-2}-1}}$ sono distinte e quindi devono coincidere con tutte le $2^{\alpha-2}$ classi resto in $U(\mathbb{Z}/2^\alpha\mathbb{Z})$ rappresentate da interi congrui a 1 modulo 4, cioè gli elementi di $(1+4\mathbb{Z})/2^\alpha\mathbb{Z}$. Anche i loro opposti sono distinti e quindi devono coincidere con tutte le classi resto rappresentate da interi congrui a -1 modulo 4. Dunque ogni elemento di $U(\mathbb{Z}/2^\alpha\mathbb{Z})$ si scrive in modo unico come $(\overline{-1})^r \cdot \overline{5^s}$ con $0 \leq r < 2$ e $0 \leq s < 2^{\alpha-2}$. Ciò è equivalente al nostro obiettivo. ⁹ \square

DIMOSTRAZIONE ALTERNATIVA PER p DISPARI. Di nuovo, anticipiamo il fatto che $U(\mathbb{Z}/p\mathbb{Z})$ è ciclico, e sia g un *generatore modulo p* (detto anche una *radice primitiva modulo p*), cioè un intero la cui classe resto genera $U(\mathbb{Z}/p\mathbb{Z})$, cioè ha ordine $p-1$. Ne segue subito che l'ordine di g modulo p^α è multiplo di $p-1$. Analogamente, dato che $g(1+p)$ coincide con g modulo p , anche l'ordine di $g(1+p)$ modulo p^α è multiplo di $p-1$. Dunque entrambi gli ordini (modulo p^α) sono multipli di $(p-1)$ e divisori di $(p-1)p^{\alpha-1}$ (che è l'ordine di $U(\mathbb{Z}/p^\alpha\mathbb{Z})$). Affermo che almeno uno fra g e $g(1+p)$ è un generatore modulo p^α . Infatti, se per assurdo entrambi gli ordini dividessero $(p-1)p^{\alpha-2}$, allora anche l'ordine del loro quoziente $(p+1)$ dividerebbe $(p-1)p^{\alpha-2}$, in contraddizione con il Lemma. \square

OSSERVAZIONI. (1) Entrambe le dimostrazioni sono costruttive (cioè producono esplicitamente un generatore modulo p^α , per p dispari), purché abbiamo a disposizione una radice primitiva modulo p . Un metodo efficiente (probabilistico) per trovare quest'ultima è implicito nella dimostrazione costruttiva che \mathbb{F}_p^* è ciclico, che si trova all'inizio del Capitolo 2.

(2) Se g è una radice primitiva modulo p^2 , per p dispari, allora g è automaticamente una radice primitiva modulo p^α , per ogni $\alpha \geq 2$. Infatti da un lato l'ordine di g modulo p^α sarà multiplo di $p-1$. Dall'altro, avendo

⁹Un modo equivalente ma più formale di farlo è il seguente. Poiché $\overline{-1} \notin \langle \overline{5} \rangle$, i due sottogruppi $\langle \overline{-1} \rangle$ e $\langle \overline{5} \rangle$ di $U(\mathbb{Z}/2^\alpha\mathbb{Z})$ hanno intersezione 1, e quindi il loro prodotto è diretto. Per questioni di ordine, esso coincide con l'intero gruppo $U(\mathbb{Z}/2^\alpha\mathbb{Z})$.

g^{p-1} ordine p modulo p^2 , avremo $g^{p-1} = 1 + ap$ con a intero primo con p . Grazie al Lemma 1.19, g^{p-1} ha ordine $p^{\alpha-1}$ modulo p^α . Ma l'ordine di g^{p-1} divide quello di g , e quindi quest'ultimo deve essere multiplo di $p^{\alpha-1}$.

- (3) Il metodo per trovare un generatore modulo p^α suggerito dalla seconda dimostrazione, per p dispari, è in pratica piú efficiente di quello dato dalla prima. Alla luce anche dell'osservazione (2), una volta ottenuta una radice primitiva g modulo p , basta verificare se $g^{p-1} \not\equiv 1 \pmod{p^2}$. In caso affermativo g è una radice primitiva modulo p^α per ogni $\alpha \geq 2$. In caso contrario $g(1+p)$ lo è (o anche $g+p$, se preferiamo, come si mostra in modo analogo).
- (4) Il fatto, oggetto dell'osservazione (2), che una radice primitiva modulo p^2 (per p dispari) è automaticamente una radice primitiva modulo p^α , si può vedere anche nel seguente modo diretto, indipendente dal Lemma 1.19.

Possiamo supporre $\alpha \geq 2$. Useremo il fatto che $U(\mathbb{Z}/p^\alpha\mathbb{Z})$ è ciclico. L'omomorfismo $\psi : U(\mathbb{Z}/p^\alpha\mathbb{Z}) \rightarrow U(\mathbb{Z}/p^2\mathbb{Z})$ dato dalla riduzione modulo p^2 (cioè la mappa $x + p^\alpha\mathbb{Z} \mapsto x + p^2\mathbb{Z}$), essendo suriettivo, manda generatori del primo gruppo in generatori del secondo. Quindi se \mathcal{G} indica l'insieme dei generatori del secondo gruppo (cioè il sottoinsieme di $U(\mathbb{Z}/p^2\mathbb{Z})$ costituito dagli elementi di ordine pari a $|U(\mathbb{Z}/p^2\mathbb{Z})| = p(p-1)$), ogni generatore di $U(\mathbb{Z}/p^\alpha\mathbb{Z})$ appartiene alla controimmagine $\psi^{-1}(\mathcal{G})$. Se mostriamo che tale controimmagine ha tanti elementi quanti i generatori di $U(\mathbb{Z}/p^\alpha\mathbb{Z})$, ne concluderemo che questi ultimi sono esattamente *tutti* gli elementi di $\psi^{-1}(\mathcal{G})$; in altre parole, che un intero (necessariamente primo con p) è una radice primitiva modulo p^2 se e solo se è una radice primitiva modulo p^α .

Ora ricordate che il numero di generatori di un gruppo ciclico è dato dalla funzione di Eulero del suo ordine. Quindi il numero di generatori di $U(\mathbb{Z}/p^\alpha\mathbb{Z})$ è $\varphi(\varphi(p^\alpha)) = \varphi((p-1)p^{\alpha-1}) = \varphi(p-1) \cdot (p-1)p^{\alpha-2}$ (una delle rare occasioni in cui serve comporre la funzione di Eulero con se stessa). D'altra parte, essendo ogni omomorfismo ψ una mappa $|\ker(\psi)|$ -a-uno, avremo ^{10 11}

$$|\psi^{-1}(\mathcal{G})| = |\mathcal{G}| \cdot |\ker(\psi)| = \varphi(\varphi(p^2)) \cdot p^{\alpha-2} = \varphi(\varphi(p^\alpha)).$$

ESERCIZIO 1.21. Ricordando dal Teorema cinese dei resti che il prodotto di due gruppi ciclici di ordini coprimi è ciclico (e ovviamente ha ordine il prodotto degli ordini), mostrate che $U(\mathbb{Z}/104\mathbb{Z}) \cong U(\mathbb{Z}/105\mathbb{Z})$.

¹⁰Se volete confrontare con un ragionamento simile ma in un contesto leggermente piú semplice, date un'occhiata alla Sottosezione 2.3.3, dove mostreremo che radici quadrate modulo un primo dispari p si possono sempre sollevare a radici quadrate modulo p^α .

¹¹Notate cosa fallirebbe se con la stessa idea volessi invece sollevare un generatore modulo p ad uno modulo p^2 , quindi usando l'omomorfismo $U(\mathbb{Z}/p^2\mathbb{Z}) \rightarrow U(\mathbb{Z}/p\mathbb{Z})$: il primo gruppo ha $\varphi(p-1) \cdot (p-1)$ generatori, che sono meno delle $\varphi(p-1) \cdot p$ controimmagini dei generatori del secondo gruppo. Questo ragionamento comunque ci dice qualcosa: una radice primitiva random modulo p ha probabilità $(p-1)/p = 1 - 1/p$ (quindi alta per p grande) di essere anche una radice primitiva modulo p^2 (e quindi anche modulo p^α).

COROLLARIO 1.22. *L'esponente di $U(\mathbb{Z}/m\mathbb{Z})$, dove $m = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_r^{\alpha_r}$ con $p_1 = 2 < p_2 < \dots < p_r$ (cioè il piú piccolo intero positivo e tale che $a^e \equiv 1 \pmod{m}$ per ogni a primo con m), coincide con il minimo comune multiplo di $\varphi(p_1^{\alpha_1}), \dots, \varphi(p_r^{\alpha_r})$ se m è dispari oppure il doppio o il quadruplo di un numero dispari (cioè se $\alpha_1 \leq 2$), altrimenti coincide con il minimo comune multiplo di $\varphi(p_1^{\alpha_1})/2, \varphi(p_2^{\alpha_2}), \dots, \varphi(p_r^{\alpha_r})$.*

In particolare, esiste una radice primitiva modulo m (cioè un intero a tale che $a + m\mathbb{Z}$ è un generatore di $U(\mathbb{Z}/m\mathbb{Z})$) se e solo se m è della forma $2, 4, p^\alpha$, o $2p^\alpha$, dove p è un primo dispari.

DIMOSTRAZIONE. Dal fatto che $U(\mathbb{Z}/m\mathbb{Z}) \cong U(\mathbb{Z}/p_1^{\alpha_1}\mathbb{Z}) \times \dots \times U(\mathbb{Z}/p_r^{\alpha_r}\mathbb{Z})$, segue subito che l'esponente di $U(\mathbb{Z}/m\mathbb{Z})$ è il minimo comune multiplo degli esponenti dei vari fattori del prodotto diretto. Grazie alla Proposizione 1.20, l'esponente di $U(\mathbb{Z}/p^\alpha\mathbb{Z})$ coincide con il suo ordine $\varphi(p^\alpha)$ se p è dispari o $\alpha \leq 2$ (essendo il gruppo ciclico in tali casi), con $\varphi(2^\alpha)/2$ altrimenti.

Per la seconda parte basta notare che un prodotto diretto di gruppi è ciclico se e solo se tutti i fattori sono ciclici e hanno ordini a due a due coprimi. ¹² \square

ESERCIZIO 1.23. Se p è un primo dispari, ecco come ricavare un generatore di $U(\mathbb{Z}/2p^\alpha\mathbb{Z})$ da uno di $U(\mathbb{Z}/p^\alpha\mathbb{Z})$. Se l'intero g è un generatore modulo p^α , mostrate che g o $g + p^\alpha$ (e precisamente quello dispari dei due) è un generatore modulo $2p^\alpha$.

OSSERVAZIONI. (1) L'esponente di $U(\mathbb{Z}/m\mathbb{Z})$ è detto talvolta la *funzione di Carmichael* di m , ed è indicato con $\lambda(m)$. La corrispondente generalizzazione del Teorema di Eulero-Fermat, $a^{\lambda(m)} \equiv 1 \pmod{m}$ se $(a, m) = 1$, è anche detta il *Teorema di Carmichael*.

(2) Per il metodo di crittografia RSA avrà particolare importanza il caso $\lambda(pq) = (p-1)(q-1)/(p-1, q-1)$, se p e q sono primi (dispari, essendo grandi) distinti. Vedremo che i primi p e q per il metodo RSA andranno scelti in modo che $(p-1, q-1)$ sia piccolo. Naturalmente per trattare questo caso $m = pq$ (cosí come, piú in generale, il caso in cui m è prodotto di primi distinti) non serve il Lemma 1.19, bastano il teorema cinese dei resti e il fatto che \mathbb{F}_p^* è ciclico.

1.3.3. L'ordine del prodotto di due elementi di un gruppo. Ripassiamo qualche fatto elementare di algebra che è bene conoscere (ed abbiamo già usato, ad esempio, nella prima dimostrazione della Proposizione 1.20).

Anzitutto ricordiamo la formula $|g^i| = |g|/(i, |g|)$ per l'ordine di una potenza g^i di un elemento g . (La notazione $|g|$ per l'ordine di un elemento g è giustificata dal fatto che $|g| = |\langle g \rangle|$.) Una conseguenza importante di questa formula è che il numero di generatori di un gruppo ciclico G di ordine n è $\varphi(n)$. Infatti se

¹²Se il prodotto diretto è ciclico, ciascuno dei fattori lo sarà, essendo isomorfo ad un sottogruppo. Se i fattori sono ciclici di ordini a due a due coprimi, il Teorema cinese dei resti mostra che il prodotto diretto è ciclico. Se due dei fattori ciclici hanno ordini non coprimi, diciamo con massimo comun divisore $d > 1$, il prodotto diretto avrà almeno d^2 elementi di ordine che divide d , e quindi non potrà essere ciclico.

$G = \langle g \rangle = \{g^0 = 1, g, g^2, g^3, \dots, g^{n-1}\}$, un suo elemento g^i genera G esattamente se $|g| = n$, e quindi il numero di generatori di G coincide con il numero di interi da 0 a $n - 1$ che sono primi con n .

Ora supponiamo di avere due elementi g ed h di un gruppo G , entrambi di ordine finito $|g|$ ed $|h|$, e di volerne dedurre qualche informazione sull'ordine del prodotto gh . Anzitutto, se $gh \neq hg$ non possiamo dedurre niente, come mostrano facili esempi, quindi supponiamo che g ed h commutino, cioè che $gh = hg$. Allora l'ordine di gh certamente divide il minimo comune multiplo degli ordini di g ed h . Infatti $(gh)^i = g^i \cdot h^i$ per ogni intero i , e quindi $(gh)^n = g^n \cdot h^n = 1 \cdot 1 = 1$ se $|g| \mid n$ e $|h| \mid n$.

In generale, l'ordine di gh può essere più piccolo di tale minimo comune multiplo, si pensi al caso limite in cui $h = g^{-1}$. Tuttavia, se $(|g|, |h|) = 1$ allora vale $|gh| = |g| \cdot |h|$ (e quindi $|gh|$ coincide con il minimo comune multiplo di $|g|$ ed $|h|$ in questo caso). Per vederlo, diciamo n l'ordine di gh , e quindi $1 = (gh)^n = g^n h^n$. Ma allora l'ordine di $g^n = h^{-n}$ divide sia $|g|$ che $|h|$, e quindi vale 1. Perciò $g^n = 1$, da cui $|g| \mid n$, ed analogamente $|h| \mid n$. Dunque n è un multiplo comune di $|g|$ ed $|h|$, ma già sapevamo che n divide il loro minimo comune multiplo, e quindi coincide con esso.

Una conseguenza di questo fatto è che ogni gruppo abeliano finito ha un elemento di ordine uguale al suo esponente.

Su di esso baseremo inoltre la dimostrazione del seguente fatto molto più generale.

LEMMA 1.24. *Siano g ed h elementi di ordine finito di un gruppo G , con $gh = hg$. Supponiamo che, per ogni primo p che divide $|g| \cdot |h|$, la potenza massima di p che divide $|g|$ sia diversa dalla potenza massima di p che divide $|h|$. Allora l'ordine del prodotto gh è il minimo comune multiplo degli ordini di g ed h .*

Per dimostrare il lemma abbiamo bisogno di un risultato intermedio, anch'esso di interesse.

LEMMA 1.25. *Se $|g| = m \cdot n$ con $(m, n) = 1$, allora esistono unici $h, k \in \langle g \rangle$ con $g = hk$, $|h| = m$ e $|k| = n$.*

Spesso si scrive $g_m := h$ e $g_n := k$.

DIMOSTRAZIONE. Esistono interi a e b tali che $am + bn = 1$, e quindi $g = hk$ con $h = (g^b)^n$ di ordine un divisore di m , e $k = (g^a)^m$ di ordine un divisore di n . D'altra parte sappiamo che $|hk|$ divide il minimo comune multiplo di $|h|$ e $|k|$, e quindi questi devono coincidere con m ed n . L'unicità si vede con un ragionamento già fatto poche righe fa, quindi la lascio per esercizio.

Alternativamente, dato che gruppi ciclici dello stesso ordine sono isomorfi, possiamo prendere $\langle g \rangle = \mathbb{Z}/mn\mathbb{Z}$ (come gruppo additivo). Grazie al teorema cinese dei resti quest'ultimo è isomorfo a $\mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$. Ogni elemento di questo prodotto diretto è chiaramente esprimibile, ed in modo unico, come somma di un elemento di ordine un divisore di m (che deve avere la forma $(*, 0)$) e di un elemento di ordine un divisore di n . \square

DIMOSTRAZIONE DEL LEMMA 1.24. Grazie al lemma precedente, se $|g| = p_1^{\alpha_1} \cdots p_r^{\alpha_r}$ (con $p_1 < \cdots < p_r$), allora $g = g_1 \cdots g_r$ con $g_i \in \langle g \rangle$ e $|g^i| = p_i^{\alpha_i}$. Analogamente, se $|h| = p_1^{\beta_1} \cdots p_r^{\beta_r}$ (con gli stessi primi di prima, come possiamo assumere), allora $h = h_1 \cdots h_r$ con $h_i \in \langle h \rangle$ e $|h^i| = p_i^{\beta_i}$. Perciò

$$|gh| = |(g_1 h_1) \cdots (g_r h_r)| = |g_1 h_1| \cdots |g_r h_r|.$$

Per ipotesi vale $\alpha_i \neq \beta_i$, diciamo $\alpha_i < \beta_i$ per un certo i . Chiaramente $|g_i h_i| \mid p_i^{\beta_i}$. Viceversa, $p_i^{\beta_i} = |h_i| = |g_i^{-1}(g_i h_i)|$ divide il minimo comune multiplo di $|g_i^{-1}| = p_i^{\alpha_i}$ e $|g_i h_i|$, che è chiaramente il secondo dei due. \square

1.3.4. Il metodo dei quadrati ripetuti. Per calcolare efficientemente una potenza $b^n \pmod{m}$ (dove assumiamo $0 \leq b < m$) una prima accortezza è quella di ridurre modulo m ogni volta che si è eseguita una moltiplicazione. Questo richiede $O(\log^2 m)$ operazioni bit ogni volta, ma mantiene i numeri da moltiplicare inferiori a m , permettendo di calcolare in $O(n \cdot \log^2 m)$ operazioni bit piuttosto che le $O(n^2 \cdot \log^2 m)$ operazioni bit richieste se riducessi modulo m soltanto una volta alla fine.

Una drastica riduzione del tempo si ottiene con il metodo dei quadrati ripetuti (o, forse meglio, *eleva al quadrato e moltiplica*), portando la stima da non polinomiale (in $\log n$ e $\log m$) a polinomiale. Se n ha k bit, in notazione binaria sarà

$$n = \sum_{i=0}^{k-1} n_i 2^i.$$

Ora calcoliamo le potenze $b^{2^i} \pmod{m}$ per $i = 0, 1, \dots, k-1$, ottenendo ciascuna come quadrato della precedente e riducendo subito modulo m (ciascun passo richiede $O(\log^2 m)$ operazioni bit quindi complessivamente richiede $O(\log n \cdot \log^2 m)$ operazioni bit). Infine moltiplichiamo fra loro, riducendo ogni volta modulo m , quelle fra le potenze $b^{2^i} \pmod{m}$ ottenute per cui vale $n_i = 1$. Anche questo si fa in $O(\log n \cdot \log^2 m)$ operazioni bit.

PROPOSIZIONE 1.26. Tempo $(b^n \pmod{m}) = O(\log n \cdot \log^2 m)$.

OSSERVAZIONI. (1) Si potrebbe pensare di rimpiazzare l'esponente n con il suo resto modulo $\varphi(m)$, ma per poterlo fare serve sostanzialmente conoscere la fattorizzazione di m . Se questa è disponibile, o se si conosce $\varphi(m)$ (e se $(b, m) = 1$), la stima scende a $O(\log n \cdot \log m + \log^3 m)$ (il primo addendo è per ridurre n modulo $\varphi(m)$).

(2) Lavorando nell'aritmetica ordinaria anziché in quella modulare il metodo dei quadrati ripetuti fa risparmiare ben poco, soltanto un misero fattore costante $\frac{5}{6}$. Infatti per calcolare la potenza n -esima di un intero b ci vogliono $\frac{1}{2}n(n-1)(\log_2 b)^2$ operazioni bit col metodo ordinario (questo è facile), mentre ne bastano $\frac{5}{12}n^2(\log_2 b)^2$ con quello dei quadrati ripetuti. (Quindi in entrambi i casi la stima è $O(n^2 \log b)$.)

Benché il fatto non abbia rilevanza pratica, verifichiamo ora la seconda stima per puro esercizio (la prima è immediata). Consideriamo il caso peggiore in cui $n = 2^k - 1$, ed iniziamo stimando il costo del calcolo dei quadrati: elevare al quadrato ciascuno dei numeri $b, b^2, \dots, b^{2^{k-2}}$ costa rispettivamente $(\log_2 b)^2, (2 \log_2 b)^2, (2^2 \log_2 b)^2, \dots, (2^{k-2} \log_2 b)^2$ operazioni bit, e quindi, complessivamente, non supera

$$\begin{aligned} 2^{2(k-2)} \left(1 + \frac{1}{4} + \frac{1}{16} + \dots \right) (\log_2 b)^2 &\leq 2^{2(k-2)} \cdot \frac{4}{3} (\log_2 b)^2 \\ &< \frac{(n+1)^2}{12} (\log_2 b)^2 \end{aligned}$$

operazioni bit, cioè circa $\frac{1}{12} n^2 (\log_2 b)^2$, se trascuriamo gli addendi meno significativi. Vediamo ora il costo delle moltiplicazioni: moltiplicare b per b^2 , il risultato per b^4 , il risultato per b^8, \dots , il risultato per $b^{2^{k-1}}$ costa rispettivamente $(\log_2 b) \cdot 2(\log_2 b), 3(\log_2 b) \cdot 4(\log_2 b), 7(\log_2 b) \cdot 8(\log_2 b), \dots$ operazioni bit, e quindi meno di, rispettivamente, $2^2 (\log_2 b)^2, 4^2 (\log_2 b)^2, \dots, (2^{k-1})^2 (\log_2 b)^2$ operazioni bit; dunque, complessivamente, il quadruplo della stima trovata per il calcolo dei quadrati, vale a dire $\frac{1}{3} n^2 (\log_2 b)^2$. Sommando le due stime trovate, si giunge a $\frac{5}{12} n^2 (\log_2 b)^2$.

Ecco uno schema conveniente per eseguire a mano uno dei possibili algoritmi per eseguire a mano il metodo dei quadrati ripetuti. Il metodo si può applicare più in generale per calcolare una potenza b^n , con n intero positivo, dove b è un elemento di un qualsiasi monoide, al posto del monoide moltiplicativo $(\mathbb{Z}/m\mathbb{Z}, \cdot)$.

Nella prima tabella si calcola l'espansione binaria dell'esponente n , come descritto in precedenza. Dunque si divide ciascun numero della prima colonna per 2 e si mette il resto a destra di esso ed il quoziente sotto di esso, continuando così fino ad ottenere quoziente zero (e da quel punto in poi saranno zero tutti i resti e tutti i quozienti). In tal modo otteniamo le cifre dell'espansione binaria $n = \sum_{i=0}^{k-1} d_i \cdot 2^i$ di n a partire dalla meno significativa, cioè nell'ordine inverso rispetto alla scrittura naturale $(d_{k-1}, d_{k-2}, \dots, d_1, d_0)_2$. Come abbiamo già visto in precedenza, la giustificazione di questa affermazione è data dalla formula

$$n = (\dots((d_{k-1} \cdot 2 + d_{k-2}) \cdot 2 + d_{k-3}) \cdot 2 + \dots + d_1) \cdot 2 + d_0.$$

Questa osservazione è importante anche in un altro contesto: la stessa idea fornisce un modo più efficiente rispetto a quello più ovvio per calcolare il valore $f(\xi)$ di un polinomio $f(x)$ su un elemento ξ .

Le cifre binarie di n compaiono poi rovesciate (e quindi nell'ordine naturale, partendo dalla più significativa) nella prima colonna della seconda tabella. Nella seconda tabella di quest'ultima si pone l'elemento neutro 1 del monoide in cima (una riga sopra rispetto alla cifra binaria più significativa d_{k-1}), quindi l'operazione base da eseguire ripetutamente è mettere in ciascuna entrata della seconda colonna il quadrato dell'elemento immediatamente sopra, moltiplicato per 1 o a a seconda che la cifra nella posizione a sinistra sia 0 o 1. Alla fine (cioè concluse le cifre binarie di n), l'ultimo elemento della seconda colonna contiene b^n .

n	$(n)_2$	$(n)_2$	$c_0 := 1$ (nel monoide M)
$(n - d_0)/2$	d_0	d_{k-1}	$c_1 := c_0^2 \cdot b^{d_{k-1}}$
$(n - d_0 - d_1 \cdot 2)/2^2$	d_1	d_{k-2}	$c_2 := c_1^2 \cdot b^{d_{k-2}}$
\vdots	\vdots	\vdots	\vdots
$d_{k-1} \cdot 2 + d_{k-2}$	\vdots	d_1	$c_{k-1} := c_{k-2}^2 \cdot b^{d_1}$
d_{k-1}	d_{k-2}	d_0	$c_k := c_{k-1}^2 \cdot b^{d_0}$
0	d_{k-1}		

A questo punto $b^n = c_k$.

Si noti che prendendo come M il gruppo *additivo* degli interi e $b = 1$ (facendo attenzione a interpretare la notazione in modo additivo, cioè $c_0 := 0$, $c_1 := 2c_0 + d_{k-1}$, ecc.), la seconda tabella è uno schema per l'algoritmo per "convertire" n da binario a decimale (o meglio, alla forma in cui decidiamo di manipolare i numeri interi), ed infatti il risultato finale è $c_k = n \cdot 1 = n$ (la n -esima potenza in un gruppo additivo è il multiplo n -esimo).

In un esempio concreto, calcoliamo $\bar{3}^{90}$ nel monoide moltiplicativo dell'anello $\mathbb{Z}/91\mathbb{Z}$, ovvero calcoliamo $3^{90} \pmod{91}$. L'esempio non è casuale, infatti lo utilizzeremo nell'ultimo Capitolo di queste Note quando studieremo i test di primalità.

$(90)_2$	$(90)_2$	1 (calcoli modulo 91)
90	0	$1^2 \cdot 3 = 3$
45	1	$3^2 = 9$
22	0	$9^2 \cdot 3 = 61 \equiv -30$
11	1	$(-30)^2 \cdot 3 = 9 \cdot 100 \cdot 3 \equiv 9^2 \cdot 3 \equiv -30$
5	1	$(-30)^2 \cdot 3 \equiv -10$
2	0	$(-10)^2 \cdot 3 \equiv 27$
1	1	$(27)^2 = 3^6 = 9^3 = 81 \cdot 9 \equiv (-10) \cdot 9 \equiv 1$
0		

1.3.5. Una proprietà della funzione di Eulero. La seguente proprietà della funzione di Eulero è utile in certe situazioni. Ad esempio, la useremo all'inizio del prossimo capitolo per dare una dimostrazione che ogni sottogruppo moltiplicativo finito di un campo è ciclico.

PROPOSIZIONE 1.27.
$$\sum_{d|n} \varphi(d) = n.$$

PRIMA DIMOSTRAZIONE. Partizioniamo l'insieme $\{0, 1, \dots, n-1\}$ a seconda del massimo comune divisore con n , cioè mettiamo j nell'insieme S_d se e solo se $(j, n) = d$. Gli S_d formano una partizione di $\{0, 1, \dots, n-1\}$, e S_d ha ordine $\varphi\left(\frac{n}{d}\right)$, in quanto S_d consiste dei $\varphi\left(\frac{n}{d}\right)$ valori di $j = id$ per cui $\left(i, \frac{n}{d}\right) = 1$.

Quindi

$$n = \sum_{d|n} \varphi\left(\frac{n}{d}\right) = \sum_{d|n} \varphi(d)$$

□

SECONDA DIMOSTRAZIONE. Notiamo anzitutto che $\varphi(n)$ è anche il numero di generatori di un gruppo ciclico G di ordine n (è chiaro se prendiamo $\mathbb{Z}/n\mathbb{Z}$ come tale gruppo ciclico), cioè il numero di elementi del gruppo che hanno ordine n . Ciascun elemento di G ha ordine un divisore di n . D'altra parte, per ogni d che divide n il gruppo G ha esattamente un sottogruppo (ciclico) di ordine d , e quindi quindi esattamente $\varphi(d)$ elementi di ordine d , da cui la tesi. ¹³ □

TERZA DIMOSTRAZIONE. Se $n = p_1^{\alpha_1} \cdots p_r^{\alpha_r}$, i divisori di n sono i numeri $d = p_1^{\beta_1} \cdots p_r^{\beta_r}$ con $0 \leq \beta_i \leq \alpha_i$, e

$$\sum_{d|n} \varphi(d) = \sum_{\beta_1, \dots, \beta_r} \varphi(p_1^{\beta_1}) \cdots \varphi(p_r^{\beta_r}) = \prod_i [1 + \varphi(p_i) + \varphi(p_i^2) + \cdots + \varphi(p_i^{\alpha_i})],$$

grazie alla moltiplicatività di φ . Ma $1 + \varphi(p_i) + \varphi(p_i^2) + \cdots + \varphi(p_i^{\alpha_i}) = 1 + (p_i - 1) + p_i(p_i - 1) + \cdots + p_i^{\alpha_i - 1}(p_i - 1) = p_i^{\alpha_i}$, e quindi $\sum_{d|n} \varphi(d) = p_1^{\alpha_1} \cdots p_r^{\alpha_r} = n$. □

OSSERVAZIONI. (1) La formula della Proposizione, che esprime n in funzione della funzione di Eulero dei divisori di n , si può invertire tramite la *formula di inversione di Möbius* (oggetto della prossima Osservazione), ottenendo la seguente espressione per $\varphi(n)$ in termini dei divisori di n : $\varphi(n) = \sum_{d|n} \mu(d) n/d$.

Naturalmente questa è un'applicazione piuttosto banale della formula di inversione, in quanto sappiamo già come calcolare $\varphi(n)$ (sempre assumendo di conoscere i divisori primi di n). In effetti, potete verificare per esercizio che la formula per $\varphi(n)$ appena scritta equivale (valutando $\mu(n/d)$) alla formula nota $\varphi(n) = n \prod_{p \text{ primo}, p|n} (1 - 1/p)$.

(2) La *formula di inversione di Möbius*: se $g(n) = \sum_{d|n} f(d)$, allora

$$f(n) = \sum_{d|n} \mu(n/d) g(d) = \sum_{d|n} \mu(d) g(n/d)$$

(le due forme della formula sono chiaramente equivalenti, si tratta solo di un cambiamento di indice $d' = n/d$), dove $\mu : \{1, 2, 3, \dots\} \rightarrow \{0, \pm 1\}$ è la funzione di Möbius, definita da $\mu(1) = 1$, $\mu(n) = 0$ se n non è *libero da quadrati* (cioè se n non è prodotto di primi distinti), $\mu(p_1 p_2 \cdots p_r) = (-1)^r$ se p_1, p_2, \dots, p_r sono primi distinti.

La proprietà fondamentale della funzione di Möbius (che è anzi la proprietà che definisce la funzione di Möbius, in un contesto molto più

¹³Un modo concreto e intuitivo di realizzare G è come il gruppo delle radici n -esime dell'unità in \mathbb{C} ; per ogni $d | n$, $\varphi(d)$ è il numero di radici d -esime primitive dell'unità, ecc.

generale) è $\sum_{d|n} \mu(d) = \delta_{1,n}$ (cioè 1 se $n = 1$ e 0 altrimenti), mediante la quale è facile dimostrare la formula di inversione.

[Il seguente fatto vi sarà forse familiare (una versione discreta del teorema fondamentale del calcolo integrale): se f e g sono funzioni a valori reali (o piú in generale in qualsiasi gruppo commutativo) definite sui numeri naturali $\{0, 1, 2, \dots\}$ tali che $g(n) = \sum_{i=0}^n f(i)$ (ovvero $g(n) = \sum_{i \leq n} f(i)$), allora $f(0) = g(0)$ e $f(n) = g(n) - g(n-1)$ per $n > 1$. La teoria dell'inversione di Möbius nella sua forma piú generale (che esula dalla teoria dei numeri) generalizza questo fatto rimpiazzando l'insieme ordinato dei numeri naturali con un insieme parzialmente ordinato (con certe proprietà). Nel caso in cui l'insieme è quello degli interi positivi ordinati rispetto alla divisibilità, si ottiene la formula di inversione vista.]

1.4. Il teorema di Dirichlet sui primi in una progressione aritmetica

TEOREMA 1.28 (di Dirichlet, sui primi in una progressione aritmetica). *Siano m ed a interi positivi con $(a, m) = 1$. Allora esistono infiniti primi congrui ad a modulo m . Equivalentemente, la progressione aritmetica $a, a+m, a+2m, a+3m, \dots$ (di primo termine a e ragione m) contiene infiniti primi.*

Chiaramente l'ipotesi che $(a, m) = 1$ è indispensabile, in quanto tutti i membri della progressione sono multipli di (a, m) .

In una sua forma piú forte, il teorema afferma anzi che l'insieme dei primi congrui ad a modulo m ha densità $1/\varphi(m)$ nell'insieme di tutti i primi, cioè esiste il

$$\lim_{n \rightarrow \infty} \frac{\#\{p \leq n \mid p \text{ primo e } p \equiv a \pmod{m}\}}{\#\{\text{primi} \leq n\}}$$

e vale $1/\varphi(m)$.

Ecco un esempio di applicazione del Teorema di Dirichlet.

PROPOSIZIONE 1.29. *Esiste una successione di primi p_j tale che la probabilità che un elemento random $g \in \mathbb{F}_{p_j}^*$ sia un generatore tende a zero.*

DIMOSTRAZIONE. Tale probabilità vale

$$\frac{\varphi(p_j - 1)}{p_j - 1} = \prod_{r \text{ primo, } r \mid (p_j - 1)} \left(1 - \frac{1}{r}\right).$$

Nella prossima sezione (Proposizione 1.32) mostreremo che $\prod_{q \text{ primo}} \left(1 - \frac{1}{q}\right) = 0$, cioè che

$$\lim_{N \rightarrow \infty} \prod_{q \text{ primo, } q < N} \left(1 - \frac{1}{q}\right) = 0.$$

Dunque è sufficiente scegliere i primi p_j in modo che l'insieme dei primi che dividono $p_j - 1$ cresca al crescere di j fino a comprendere, prima o poi, qualsiasi primo prefissato. Ad esempio, possiamo scegliere il primo p_j in modo che $p_j \equiv 1 \pmod{j!}$. Che questo si possa fare è garantito dal Teorema di Dirichlet. \square

Una dimostrazione generale del Teorema di Dirichlet sarebbe troppo difficile per questo corso, ma ne dimostreremo qualche caso particolare. Per cominciare, il caso in cui la progressione aritmetica ha ragione $m = 2$ è ovvio: stiamo semplicemente affermando che esistono infiniti numeri primi dispari.

È anche facile mostrare che se $m > 2$ esistono infiniti primi che *non sono* congrui a 1 modulo m , procedendo in modo simile alla famosa dimostrazione di Euclide dell'esistenza di infiniti primi (riportata all'inizio della prossima sezione). Supponiamo per assurdo che p_1, p_2, \dots, p_r siano tutti i primi non congrui a 1 modulo m , e consideriamo l'intero $s = mp_1p_2 \cdots p_r - 1$, che è chiaramente congruo a -1 modulo m . Tutti i suoi fattori primi sono diversi da p_1, p_2, \dots, p_r , e quindi sono per ipotesi congrui a 1 modulo m . Quindi anche il loro prodotto s lo è, contraddizione.

In particolare, questo risultato implica la validità del Teorema di Dirichlet per $a = -1$ ed $m = 3, 4, 6$. Altri casi particolari del Teorema si possono dimostrare in modo simile. (Ad esempio, il caso $m = 8$ (con i quattro sottocasi $a = -3, -1, 1, 3$) si potrebbe dimostrare con un ragionamento simile e la legge di reciprocità quadratica di Gauss, si veda il foglio di esercizi per la settima ed ottava settimana dell'A.A. 98/99.) Dimostreremo qui il caso dove $a = 1$ ed m è primo. Ci serve il lemma seguente, che enunciamo e dimostriamo in maggiore generalità per poterlo anche usare altrove.

LEMMA 1.30. *Siano a ed n interi maggiori di 1. Se un primo p divide $a^n - 1$, allora o $p \mid a^d - 1$ per un divisore proprio d di n , oppure $p \equiv 1 \pmod{n}$.*

Notate che se p ed n sono dispari, nel secondo caso si può concludere che $p \equiv 1 \pmod{2n}$.

DIMOSTRAZIONE. Sia p un primo che divide $a^n - 1$. Allora l'ordine d di a modulo p divide n . In ogni caso $p \mid a^d - 1$, quindi se $d < n$ vale la prima conclusione. Altrimenti concludiamo che il gruppo $U(\mathbb{Z}/p\mathbb{Z})$ contiene un elemento (la classe resto di a) di ordine n , e quindi n divide $p - 1$ grazie al Teorema di Lagrange. \square

PROPOSIZIONE 1.31. *Sia q un primo. Allora ogni divisore primo p di $1 + a + a^2 + \cdots + a^{q-1}$ soddisfa $p \equiv 1 \pmod{q}$ o $p = q$. Si può dedurre che esistono infiniti primi $p \equiv 1 \pmod{q}$.*

DIMOSTRAZIONE. Essendo $(1 + a + a^2 + \cdots + a^{q-1})(a - 1) = a^q - 1$, grazie al lemma precedente abbiamo che o $a \equiv 1 \pmod{p}$, oppure $p \equiv 1 \pmod{q}$. Nel primo caso abbiamo $1 + a + a^2 + \cdots + a^{q-1} \equiv q \pmod{p}$, e quindi $p = q$.

Per dimostrare la seconda asserzione supponiamo per assurdo che p_1, p_2, \dots, p_r siano tutti i primi congrui a 1 modulo q , e poniamo $a = qp_1p_2 \cdots p_r$. Ora, ogni divisore primo p di $1 + a + a^2 + \cdots + a^{q-1}$ è diverso da q , e quindi è congruo a 1 modulo q grazie alla prima parte, ma è anche diverso da ciascuno dei primi p_1, p_2, \dots, p_r , contraddizione. \square

Il lemma che abbiamo usato sopra è utile anche in un altro contesto (che non c'entra con il Teorema di Dirichlet), quello della fattorizzazione di interi della

forma particolare $a^n - 1$. Ad esempio, i numeri $2^2 - 1$, $2^3 - 1$, $2^5 - 1$, $2^7 - 1$ sono primi, dei cosiddetti *primi di Mersenne*, mentre $2^{11} - 1 = 2047$ non lo è, come si potrebbe verificare rapidamente con un test di primalità probabilistico (un po' sprecato per un numero così piccolo, naturalmente). Volendo scoprirne una fattorizzazione con il metodo delle divisioni per tentativi, non serve provare a dividere per $3, 5, 7, \dots$, perché grazie al lemma i divisori primi p di 2047 devono essere congrui a 1 modulo 11, e quindi anche modulo 22 essendo dispari. Perciò basta limitarsi a provare a dividere 2047 per 23 (in quanto 47, il prossimo primo congruo a 1 modulo 22, già supera $\sqrt{2047}$), ed infatti si trova che $2047 = 23 \cdot 89$.

1.5. Cenni sulla distribuzione dei numeri primi

Come sapete, esistono infiniti numeri primi: questo è il *Secondo teorema di Euclide* (o *Teorema fondamentale dell'Aritmetica*).¹⁴ La dimostrazione data da Euclide (ne esistono tante altre) è la seguente: l'intero $s = 2 \cdot 3 \cdot 5 \cdots p + 1$ (ma anche $2 \cdot 3 \cdot 5 \cdots p - 1$ andrebbe bene, per $p > 2$), dove sono inclusi nel prodotto tutti i primi nell'ordine fino al primo p , non è divisibile per alcuno dei primi $2, 3, \dots, p$, quindi o s è esso stesso primo, o è divisibile per un primo compreso fra p ed s ; in ogni caso, esiste un primo maggiore di p , il che prova il teorema.

Se p è l' n -esimo primo p_n ed s è definito come sopra, avremo $q < p_n^n + 1$, e quindi, a maggior ragione, $p_{n+1} < p_n^n + 1$. Quindi già la dimostrazione originale di Euclide permetterebbe di dare una stima superiore alla crescita della successione p_n (cioè di mostrare che la funzione n -esimo primo non cresce troppo rapidamente, cioè che i primi non sono solo infiniti, ma sono "tanti"). Sarebbe comunque una stima ridicolmente alta: ora otterremo una stima molto più vicina al vero.

PROPOSIZIONE 1.32. Vale $\prod_{p \text{ primo}, p < N} \frac{1}{1 - 1/p} > \log N$ per ogni $N > 2$. In particolare, il prodotto $\prod_{p \text{ primo}, p < N} (1 - 1/p)^{-1}$ diverge a $+\infty$.

DIMOSTRAZIONE.

$$\prod_{p \text{ primo}, p < N} \left(1 - \frac{1}{p}\right)^{-1} = \prod_{p \text{ primo}, p < N} \left(1 + \frac{1}{p} + \frac{1}{p^2} + \frac{1}{p^3} + \dots\right) > \sum_{n=1}^{N-1} \frac{1}{n},$$

in quanto ciascun termine $\frac{1}{n}$ della sommatoria fattorizza in modo unico come prodotto $p_1^{-1} p_2^{-1} \dots p_r^{-1}$ con p_1, \dots, p_r primi minori di N . Inoltre

$$\sum_{n=1}^{N-1} \frac{1}{n} \geq \int_1^N \frac{dx}{x} = \log N.$$

¹⁴Il *Primo teorema di Euclide* afferma invece che se p è un (intero) primo (o meglio *irriducibile*, nella terminologia moderna), allora da $p \mid ab$ segue che $p \mid a$ o $p \mid b$ (e questa proprietà è la definizione moderna di *primo*).

(Alternativamente, si può mostrare che $\sum_{n=1}^{N-1} \frac{1}{n} \geq \log N$ per induzione su N :

$$\sum_{n=1}^{N-1} \frac{1}{n} \geq \log(N-1) + \frac{1}{N-1} \geq \log N.$$

□

OSSERVAZIONI. (1) Dal fatto che $\prod_{p < N, p \text{ primo}} (1 - 1/p)$ converge a zero deduciamo che esistono interi n con $\varphi(n)/n$ arbitrariamente vicino a zero. Infatti, $\varphi(n)/n = \prod_{p \text{ primo}, p|n} (1 - 1/p)$. In particolare, esistono gruppi ciclici in cui i generatori sono una frazione arbitrariamente piccola di tutti gli elementi.

(2) Formalmente abbiamo

$$\prod_{p \text{ primo}} \frac{1}{1 - p^{-1}} = \sum_{n=1}^{\infty} \frac{1}{n}$$

anche se naturalmente sia il prodotto infinito che la serie divergono a $+\infty$. Ma più in generale potremmo mostrare in modo analogo che vale

$$\prod_{p \text{ primo}} \frac{1}{1 - p^{-s}} = \sum_{n=1}^{\infty} \frac{1}{n^s} =: \zeta(s)$$

per ogni numero complesso s con parte reale maggiore di 1 (che è la condizione necessaria e sufficiente per la convergenza del secondo membro). Qui l'uguaglianza non è solo formale, ma significa che entrambi i membri convergono, ed allo stesso limite finito; per s reale e maggiore di 1 basta infatti notare che $\zeta(s)$ converge e che

$$\zeta(s) \geq \prod_{p \text{ primo}, p < N} \frac{1}{1 - p^{-s}} \geq \sum_{n < N} \frac{1}{n^s}.$$

La funzione $\zeta(s)$ è la *funzione zeta di Riemann*, e quella vista è una sua famosa fattorizzazione in prodotto infinito, la fattorizzazione di Eulero.

La Proposizione 1.32 fornisce una delle tante dimostrazioni che esistono infiniti numeri primi (altrimenti il prodotto sarebbe un prodotto finito, e quindi non potrebbe divergere), ma dice di più, come illustra il seguente Corollario.

COROLLARIO 1.33. Vale la stima $\sum_{p \text{ primo}, p < N} \frac{1}{p} > \frac{1}{2} \log \log N$. In particolare, la serie $\sum_{p \text{ primo}} \frac{1}{p}$ diverge a $+\infty$.

DIMOSTRAZIONE. Prendendo il logaritmo di entrambi i membri nella Proposizione 1.32 troviamo

$$- \sum_{p \text{ primo}, p < N} \log \left(1 - \frac{1}{p} \right) > \log \log N.$$

Poi basta notare che $-\log(1-x) < 2x$ per $0 \leq x \leq \frac{1}{2}$ (qui 2 non è la migliore costante possibile), da cui la tesi. \square

Il corollario ci dice che i numeri primi sono *tanti*. Infatti se indichiamo con p_n l' n -esimo numero primo (cioè $p_1 = 2$, $p_2 = 3$, $p_3 = 5 \dots$), il Corollario ci dice che tale successione (purché abbia una qualche regolarità, e questa è una delle cose difficili da dimostrare) cresce *abbastanza lentamente*. Infatti $\sum_{n=1}^{\infty} \frac{1}{p_n} = +\infty$ vuol dire che la successione p_n cresce in modo *quasi lineare* come funzione di n , nel senso che cresce più lentamente di ogni successione del tipo $n^{1+\epsilon}$ con $\epsilon > 0$, in quanto $\sum_{n=1}^{\infty} \frac{1}{n^{1+\epsilon}} < +\infty$ se $\epsilon > 0$ (ma, d'altra parte, crescerà almeno velocemente quanto n , essendo i primi un sottoinsieme dell'insieme degli interi positivi).

Ora abbiamo bisogno di una definizione che rappresenti stime più precise di quelle date dalla notazione O -maiuscola.

DEFINIZIONE 1.34. Per funzioni a valori reali positivi definite sugli interi positivi (o eventualmente sugli interi n maggiori di un certo n_0), scriveremo che $f(n) \sim g(n)$ (e diremo che f è *asintotica* a g) se vale $\lim_{n \rightarrow +\infty} \frac{f(n)}{g(n)} = 1$.

Notate che da $f \sim g$ segue che $f = O(g)$ e che $g = O(f)$, ma non viceversa!

ESEMPIO 1.35. Nell'esempio sulla complessità del calcolo di $n!$ abbiamo osservato che $\log(n!) = O(n \log n)$, e anche che $n \log n = O(\log(n!))$. Chiaramente da queste stime non segue affatto che $\log(n!) \sim n \log n$, e nemmeno che $\log(n!)/(n \log n)$ abbia un limite per $n \rightarrow +\infty$ (ma solo che esistono costanti positive c_1 e c_2 tali che valga $c_1 \leq \log(n!)/(n \log n) \leq c_2$ per ogni n maggiore di un certo valore). Tuttavia una stima più precisa per $\log(n!)$ si trova facilmente confrontando con degli integrali: $n \log n - n + 1 \leq \log(n!) \leq (n+1) \log(n+1) - n$. In particolare, segue che $\log(n!) \sim n \log n$. (Infatti il rapporto fra la stima superiore e quella inferiore tende a 1. Attenzione però, la loro differenza non tende a zero, anzi, non è nemmeno limitata: è soltanto $O(\log n)$.)

Esponenziando, otteniamo che $n^n e^{-n+1} \leq n! \leq (n+1)^{n+1} e^{-n}$. Tuttavia qui il rapporto fra la stima superiore e quella inferiore è soltanto $O(n)$ (per la precisione, è $\sim n$, verificatelo), e quindi non ne possiamo ottenere una stima asintotica per $n!$.

Comunque, una stima asintotica per $n!$ esiste, ed è detta l'*approssimazione di Stirling*: $n! \sim \sqrt{2\pi n} \cdot n^n \cdot e^{-n}$ (che è più o meno la media geometrica delle stime superiore ed inferiore, a parte la costante $\sqrt{2\pi}$). In [Kob94, Exercise V.3.7] trovate qualche passo in direzione dell'approssimazione di Stirling, e precisamente che ciascun membro è O dell'altro (in particolare, non si trova la costante $\sqrt{2\pi}$).

Un modo di “contare” i primi equivalente alla funzione $n \mapsto p_n$ è la funzione

$$\pi(n) := \#\{\text{primi minori di } n\}$$

(che all’occorrenza possiamo anche pensare definita per n reale anziché intero).

TEOREMA 1.36 (Teorema dei numeri primi). $\pi(n) \sim \frac{n}{\log n}$.

La dimostrazione del Teorema dei numeri primi è piuttosto difficile, e la omettiamo. (Tuttavia, se siamo disposti a dare per buone alcune cose e a sorvolare sui problemi piú delicati, una “quasi dimostrazione” diventa accessibile, vedi la prossima sezione.)

Se consideriamo $\pi(q)$ definita solo se q è primo, le due funzioni $q \mapsto \pi(q)$ e $n \mapsto p_n$ divengono l’una l’inversa dell’altra, infatti $\pi(p_n) = n$ e $p_{\pi_q} = q$. Dunque una stima per $\pi(n)$ si dovrebbe poter tradurre in una stima per p_n .

TEOREMA 1.37 (Forma equivalente del T. dei numeri primi). $p_n \sim n \log n$.

DIMOSTRAZIONE DELL’EQUIVALENZA. In generale, funzioni inverse di funzioni fra loro asintotiche non sono necessariamente fra loro asintotiche (ad esempio, $1 + \log x \sim \log x$, ma $e^{y-1} \not\sim e^y$), tuttavia sarebbe facile vedere che ciò vale se le funzioni sono polinomiali (o quasi, ad esempio ciascuna compresa fra due funzioni polinomiali, come nel nostro caso). Dunque p_n dovrebbe essere asintotica alla funzione inversa di $n/\log n$, che non è esprimibile in termini di funzioni elementari. Tuttavia tale inversa è asintotica ad un’opportuna funzione elementare, che troviamo ragionando nel modo seguente.

Se le due variabili x ed y sono legate dalla relazione $y = x/\log x$, allora $\log y = \log x - \log \log x$, quindi $\log y \sim \log x$, e dunque $x = y \log x \sim y \log y$. Pertanto la funzione inversa di $n/\log n$ è asintotica a $n \log n$.

In modo analogo si dimostra che la funzione inversa di $n \log n$ è asintotica a $n/\log n$. \square

OSSERVAZIONI (non troppo rigorose). (1) La seconda forma del Teorema dei numeri primi conferma la nostra osservazione, dedotta dall’ultimo Corollario, che p_n cresce piú lentamente di $n^{1+\epsilon}$.

(2) La seconda forma del Teorema dei numeri primi ci dice anche che la densità media dei primi nell’intervallo da 1 a x è $1/\log x$. In realtà questa è anche la densità media dei primi in un piccolo intervallo intorno ad x . (Dunque, benché ci siano piú primi all’inizio dell’intervallo che alla fine, la densità locale intorno ad x differisce poco da quella globale.)

Infatti possiamo pensare $\pi(n) = \int_1^n \delta(x) dx$, dove $\delta(x)$ è la funzione densità (in realtà δ non è una funzione, ma una *distribuzione*), quindi $\delta(n)$ è circa la derivata di $x/\log x$, che vale $(\log x - 1)/\log^2 x$, cioè circa $1/\log x$.

Conoscere questa densità è importante perché permette di stimare il numero di volte che dobbiamo applicare un test di primalità nella ricerca di un primo molto grande, ad esempio per l’utilizzo nel metodo RSA. Ad esempio, se cerchiamo un primo della grandezza intorno a 10^{100} , (adatto

qualche anno fa per l’RSA, oggi forse bisogna arrivare a 10^{150}) scegliamo un intero random m all’incirca di quella grandezza, e poi applichiamo un test di primalità a $m, m + 1, m + 2, \dots$: ci aspetteremo di scoprire un primo al massimo dopo un numero di passaggi dell’ordine di $\log m$, che è circa 230 (pochi). (Naturalmente anziché testare tutti gli interi a partire da m , converrà testare solo quelli dispari, dimezzando la stima; oppure solo quelli congrui a ± 1 modulo 6, e la stima viene divisa per tre...)

La seguente tabella mostra come la densità dei primi intorno ad x si approssimi bene con $1/\log x$.

intervallo da 10^n a $10^n + 1000$	numero di primi p	$1000/\log(10^n)$
$10^6 < p < 10^6 + 1000$	75	72,3...
$10^9 < p < 10^9 + 1000$	49	48,2...
$10^{12} < p < 10^{12} + 1000$	37	36,1...
$10^{15} < p < 10^{15} + 1000$	24	28,9...
$10^{18} < p < 10^{18} + 1000$	23	24,1...
$10^{21} < p < 10^{21} + 1000$	20	20,6...
$10^{24} < p < 10^{24} + 1000$	16	18,0...
$10^{27} < p < 10^{27} + 1000$	14	16,0...
$10^{30} < p < 10^{30} + 1000$	13	14,4...

- (3) Poiché la densità dei primi intorno ad x è circa $1/\log x$, ci possiamo aspettare che la differenza fra due primi consecutivi vicini ad x sia mediamente circa $\log x$. In effetti, questo si può ricavare anche direttamente da $p_n \sim n \log n$: la differenza $p_{n+1} - p_n$ dovrebbe essere in media $(n+1) \log(n+1) - n \log n = n(\log(n+1) - \log n) + \log(n+1)$, e il primo addendo tende a 1 per $n \rightarrow +\infty$.

Attenzione però, $p_{n+1} - p_n$ non è asintotico a $\log n$. Infatti, ad esempio, secondo una famosa congettura tale differenza vale 2 per infiniti valori di n (e in tal caso p_n e p_{n+1} sono detti *primi gemelli*, come 3 e 5, oppure 2129 e 2131); se la congettura è vera (ma ci sono altri ragionamenti per evitarla), il quoziente $(p_{n+1} - p_n)/\log n$ assume valori arbitrariamente vicini a zero, e quindi non può tendere a 1.

- (4) Usando le stime asintotiche per $\pi(n)$ e p_n possiamo stimare la funzione

$g(N) = \sum_{p \text{ primo}, p < N} \frac{1}{p}$ di cui abbiamo già dimostrato una stima dal basso nell’ultimo Corollario:

$$\begin{aligned}
 g(N) &= \sum_{n=1}^{\pi(N)-1} \frac{1}{p_n} \approx \sum_{n=2}^{\pi(N)-1} \frac{1}{n \log n} \sim \int_2^{\pi(N)} \frac{dx}{x \log x} = \\
 &= \log \log(\pi(N)) \sim \log \log \left(\frac{N}{\log N} \right) \sim \log \log(N)
 \end{aligned}$$

Vediamo così che la stima dal basso dimostrata nell’ultimo Corollario aveva la forma corretta.

[Questa appena trovata non è necessariamente una stima asintotica: dove appare il simbolo \sim si tratta veramente di stime asintotiche, ad esempio nel primo caso in quanto la differenza fra la sommatoria e l'integrale è ≤ 1 ; invece il simbolo \approx va letto qui (per noi) solo come "all'incirca", fidatevi.]

Rifacciamolo in un altro modo (ancor meno preciso perché trattiamo g come una funzione derivabile, mentre è definita solo sugli interi positivi): $1/N \sim 1/p_n \approx g(N + \log N) - g(N) \approx (\log n) \cdot g'(n)$, da cui $g'(N) \approx 1/(N \log N)$, ed infine $g(N) \approx \log \log N$.

1.5.1. Una dimostrazione del Teorema dei numeri primi. Questo argomento è tratto per ora da [CR71].

Cominciamo col fornire una stima asintotica per

$$\log(n!) = \log 2 + \log 3 + \cdots + \log n.$$

Si fa in maniera molto semplice, confrontando con un integrale. Risulta

$$O(\log n!) = n \cdot \log n,$$

o meglio

$$\lim_{n \rightarrow +\infty} \frac{\log(n!)}{n \cdot \log n} = 1.$$

Sia adesso

$$\pi(n) = \# \{ p : p \text{ è primo, } p \leq n \}.$$

Vogliamo far vedere che vale il

TEOREMA 1.38 (Teorema dei numeri primi).

$$\pi(n) \sim \frac{n}{\log n},$$

ovvero

$$\lim_{n \rightarrow +\infty} \frac{\pi(n)}{n/\log n} = 1.$$

In realtà faremo una trattazione molto informale, che richiederebbe un certo impegno per essere sistemata. L'assunzione più impegnativa è che $\pi(n)$ si possa stimare come

$$(1) \quad \pi(n) \sim \int_2^n W(x) dx,$$

per qualche funzione $W(x)$ che misura la densità dei numeri primi.

Andiamo adesso a vedere quale è la massima potenza p^k di un numero primo p che divide $n!$. Dato che i numeri divisibili per p vengono uno ogni p , quelli divisibili per p^2 vengono uno ogni p^2 , si ha subito

$$k \approx \frac{n}{p} + \frac{n}{p^2} + \frac{n}{p^3} + \cdots \approx \frac{n}{p-1}.$$

Dunque per n grande abbiamo

$$n! \approx \prod_{p \leq n} p^{n/(p-1)},$$

ove p varia sui primi, ovvero

$$\log(n!) \approx \sum_{p \leq n} \frac{n}{p-1} \cdot \log p.$$

Dunque, cambiando variabile,

$$\log x \approx \sum_{p \leq x} \frac{\log p}{p-1}.$$

Adesso otteniamo come conseguenza dell'assunzione (1) (spiegare) che questa formula può essere riscritta come

$$\log x \approx \int_2^x W(z) \frac{\log z}{z-1} dz.$$

Prendendo l'eguaglianza al posto dell'approssimazione, e derivando

$$\frac{1}{x} = W(x) \frac{\log x}{x-1},$$

ovvero

$$W(x) = \frac{x-1}{x \cdot \log x}.$$

Questa funzione non si può integrare elementarmente. Tutto dipende dall'integrale

$$\int \frac{1}{\log x} dx,$$

che ponendo $t = \log x$ si riduce all'integrale esponenziale

$$\int \frac{e^t}{t} dt.$$

Anche qui procediamo con una approssimazione, dato che per x grandi le due espressioni

$$\frac{x-1}{x \cdot \log x} = \frac{1}{\log x} - \frac{1}{x \cdot \log x} \quad \text{e} \quad f'(x) = \frac{1}{\log x} - \frac{1}{(\log x)^2},$$

ove $f(x) = x/\log x$, possono essere spacciate per quasi eguali. Dunque possiamo approssimare

$$\pi(x) = \int_2^x W(z) dz \approx \int_2^x f'(z) dz = \frac{x}{\log x} - \frac{2}{\log 2},$$

da cui

$$\pi(n) \sim \frac{n}{\log n}.$$

1.6. La moltiplicazione alla Montgomery

Si tratta di un metodo, dovuto a Peter L. Montgomery [Mon85], per fare rapidamente la moltiplicazione modulo N , evitando in buona sostanza la divisione per N . Il metodo richiede una rappresentazione non standard per le classi resto modulo N , dunque una certa quantità di *preprocessing*, per cui si presta bene ad applicazioni ove ci sia bisogno di eseguire molte operazioni del genere.

Vogliamo svolgere in modo efficiente le moltiplicazioni modulo N . Scegliamo un numero $R > N$ in modo che $(R, N) = 1$, e che i calcoli modulo R siano facili: in particolare la riduzione modulo R deve essere facile. Ad esempio R può essere una potenza di 2, se stiamo lavorando con numeri binari, tipo la *word size* della macchina su cui stiamo lavorando.

Calcoliamo una volta per tutte due numeri interi R^{-1} e N' , compresi fra 0 e N , tali che

$$RR^{-1} - NN' = 1.$$

Dunque R^{-1} è l'inverso di R modulo N .

Introduciamo ora una funzione $\text{REDC}(t)$ che calcola rapidamente la classe resto di $t \cdot R^{-1} \pmod{N}$, per $0 \leq t < RN$. L'idea è di calcolare in modo efficiente la classe resto modulo N di

$$tR^{-1} = t \cdot \frac{1 + NN'}{R}.$$

Procediamo come segue

$$\begin{aligned} m &:= (t \pmod{R}) \cdot N' \pmod{R} \\ \text{REDC}(t) &:= \frac{t + m \cdot N}{R} \pmod{N} \end{aligned}$$

Notiamo che ovviamente $0 \leq m < R$. Inoltre $mN \equiv tNN' \equiv -t \pmod{R}$, per cui $\text{REDC}(t)$ è un intero. Notiamo anche che $t + mN < 2RN$. Dunque

$$0 \leq \frac{t + m \cdot N}{R} < 2N$$

per cui la riduzione modulo N presente nella formula per $\text{REDC}(t)$ si riduce al massimo a sottrarre N una volta.

Dato che la riduzione modulo R è facile, $\text{REDC}(t)$ è un buon algoritmo. Invece i calcoli modulo R^{-1} non è detto che siano facili, per cui per calcolare $\text{BLUEC}(t) \equiv t \cdot R \pmod{N}$ conviene usare $\text{BLUEC}(t) := \text{REDC}(tR^2)$, ove $R^2 \pmod{N}$ va precalcolato.

Supponiamo ora di voler calcolare il prodotto di x e y modulo N , per $0 \leq x, y < N$. Procediamo con i seguenti passaggi:

$$\begin{aligned} (x, y) &\mapsto (xR \pmod{N}, yR \pmod{N}) \\ &\mapsto ((xR) \cdot (yR)) \cdot R^{-1} \equiv (xy)R \pmod{N} \\ &\mapsto xy \pmod{N}. \end{aligned}$$

Qui il primo passaggio si fa applicando $\text{BLUEC}(x)$ e $\text{BLUEC}(y)$, il secondo applicando $\text{REDC}((xR) \cdot (yR))$, e l'ultimo applicando di nuovo REDC .

CAPITOLO 2

Campi finiti e resti quadratici

2.1. Campi finiti

Sia F un campo finito. allora il suo *campo primo* è un campo con p elementi, il campo $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$, p primo. Quindi F è uno spazio vettoriale su \mathbb{F}_p , e perciò ha ordine $q = p^f$ una potenza di p .

PROPOSIZIONE 2.1. *Un sottogruppo finito del gruppo moltiplicativo di un campo E è ciclico.*

DIMOSTRAZIONE. Preliminarmente noto che per ogni n , nel campo E esistono al più n elementi distinti a tali che $a^n = 1$, in quanto sono le radici del polinomio $x^n - 1$. Quindi se G è un sottogruppo finito di ordine n di E^* , allora G gode della proprietà: per ogni $d \mid n$, esistono in G al più d elementi a tali che $a^d = 1$. Mostro ora che un gruppo finito G con tale proprietà è necessariamente ciclico. Infatti se esiste $b \in G$ di ordine d , allora tutti i d elementi di $\langle b \rangle$ hanno ordine che divide d ; viceversa, grazie all'ipotesi, tutti gli $a \in G$ tali che $a^d = 1$ appartengono a $\langle b \rangle$. In particolare, tutti gli elementi di G di ordine d sono generatori di $\langle b \rangle$.

Dunque, se indichiamo con $\psi(d)$ il numero di elementi di G che hanno ordine d , avremo che $\psi(d) \leq \varphi(d)$ per ogni $d \mid n$. Essendo $n = |G| = \sum_{d \mid n} \psi(d) \leq \sum_{d \mid n} \varphi(d) = n$, concludiamo che la disuguaglianza intermedia è un'uguaglianza, e quindi $\psi(d) = \varphi(d)$ per ogni $d \mid n$. In particolare otteniamo che $\psi(n) = \varphi(n)$, cioè G contiene $\varphi(n) \geq 0$ elementi di ordine $n = |G|$. Quindi G è ciclico. \square

DIMOSTRAZIONE COSTRUTTIVA. Sia $|G| = p_1^{n_1} \dots p_k^{n_k}$ con p_k primi distinti. Poiché $x^{|G|/p_i} - 1$ ha al più $|G|/p_i$ radici, esiste $a_i \in G$ con $a_i^{|G|/p_i} \neq 1$. Se pongo $b_i = a_i^{|G|/p_i^{n_i}}$, trovo che $b_i^{p_i^{n_i}} = 1$, ma $b_i^{p_i^{n_i-1}} \neq 1$, quindi b_i ha ordine $p_i^{n_i}$. Dunque $b_1 \dots b_k$ ha ordine $p_1^{n_1} \dots p_k^{n_k}$ (un prodotto di elementi che commutano tra loro e hanno ordini a due a due coprimi, ha ordine il prodotto dei loro ordini), e quindi genera G . \square

La seconda dimostrazione, essendo costruttiva, fornisce un metodo esplicito per trovare un generatore di G . Conviene ricordare che nei casi pratici un uso opportuno del Lemma 1.24 può abbreviare notevolmente il procedimento.

Diremo *generatore* di un campo finito \mathbb{F}_q un generatore g del suo gruppo moltiplicativo (cioè un elemento di ordine $q - 1$). Grazie alla Proposizione 2.1 ogni campo finito \mathbb{F}_q ha un generatore g . Inoltre g^j è anch'esso un generatore se e solo se $(j, q - 1) = 1$, quindi in particolare esistono in totale $\varphi(q - 1)$ generatori.

In particolare, per ogni primo p esiste un intero g tale che ogni classe resto non nulla modulo p contiene una potenza di g (e la dimostrazione costruttiva della

Proposizione 2.1 ci dice come trovare tale g). Un tale g si chiama anche una *radice primitiva modulo p* . Ad esempio, la piú piccola radice primitiva modulo p è 2 per $p = 3, 5, 11, 13, 19, 29, 37, \dots$; è 3 per $p = 7, 17, 31, 43, \dots$; è 5 per $p = 23, 47, \dots$; è 6 per $p = 41, \dots$ (Modulo 41, ad esempio, 2 ha ordine 20, 3 ha ordine 8, 5 ha ordine 20; modulo 47, invece, 2 e 3 hanno entrambi ordine 23.)

È inoltre sempre possibile scegliere un primo g come radice primitiva modulo p , (ad esempio, 11 è una radice primitiva modulo 41), grazie al Teorema di Dirichlet sui primi in una progressione aritmetica, il Teorema 1.28.

Una famosa *congettura di Artin* afferma che ogni intero a che non sia un quadrato perfetto e che sia diverso da -1 è una radice primitiva modulo infiniti primi p . (La congettura è anzi piú forte, ed afferma che tale insieme di primi, per un a fissato, ha densità positiva nell'insieme di tutti i primi, e propone un valore esplicito per tale densità.)

PROPOSIZIONE 2.2. *Se \mathbb{F}_q è un campo finito di $q = p^f$ elementi, allora esso è un campo di spezzamento su \mathbb{F}_p del polinomio $x^q - x$. Viceversa, se F è un campo di spezzamento di $x^q - x$ su \mathbb{F}_p , dove $q = p^f$, allora $|F| = q$. In particolare, per ogni potenza q di un primo p esiste un campo con q elementi, ed è unico a meno di isomorfismo (grazie all'unicità del campo di spezzamento).*

DIMOSTRAZIONE. La prima parte segue dal fatto che tutti gli elementi di \mathbb{F}_q sono radici di $x^q - x$, che quindi si spezza su \mathbb{F}_p in un prodotto di fattori lineari

$$x^q - x = \prod_{\alpha \in \mathbb{F}_q} (x - \alpha).$$

Per la seconda parte, sia V l'insieme delle radici di $x^q - x$ nel suo campo di spezzamento F su \mathbb{F}_p . Anzitutto $|V| = q$ in quanto $x^q - x$ ha radici distinte (poiché $D(x^q - x) = qx^{q-1} - 1 = -1$ è primo con $x^q - x$). Ora V è un sottomonoido moltiplicativo di \mathbb{F} , ma è anche un sottogruppo additivo (poiché $(a + b)^q = a^q + b^q$ in caratteristica p), quindi è un sottocampo di F , e dunque coincide con F . \square

Se F è un campo di caratteristica p , la mappa $\sigma : F \rightarrow F$ tale che $\sigma(a) = a^p$ è un endomorfismo di campi, iniettivo poiché $a^p = 0$ implica $a = 0$. Se F è infinito può succedere che la sua immagine F^p sia un sottocampo proprio di F . Se $F^p = F$ il campo si dice *perfetto* e σ è un automorfismo, detto l'*automorfismo di Frobenius*. In particolare questo vale ovviamente se $F = \mathbb{F}_q$ è finito (diciamo $q = p^f$). I punti fissi di σ^j sono le radici in \mathbb{F}_q del polinomio $x^{p^j} - x$, quindi in particolare l'ordine di σ (cioè il piú piccolo $j > 0$ tale che σ fissa ogni elemento di \mathbb{F}_q) è f .¹

PROPOSIZIONE 2.3. *Se $\alpha \in \mathbb{F}_q$, allora le radici del suo polinomio minimo su \mathbb{F}_p sono distinte e sono i coniugati di α su \mathbb{F}_p (cioè le immagini di α sotto automorfismi di \mathbb{F}_q , ovvero gli elementi $\sigma^j(\alpha) = \alpha^{p^j}$).*

¹In questo modo si ottengono tutti gli automorfismi di \mathbb{F}_q . Un modo di vederlo è usare un lemma di Dedekind sull'indipendenza lineare degli automorfismi, che implica che un'estensione di campi E/F di grado finito ha al piú $|E : F|$ automorfismi.

DIMOSTRAZIONE. Sia $g(x)$ il polinomio minimo di α su \mathbb{F}_p e d il suo grado, dunque $\mathbb{F}_p(\alpha) \cong \mathbb{F}_p[x]/(g(x))$ e $d = |\mathbb{F}_p(\alpha) : \mathbb{F}_p|$. Perciò $\mathbb{F}_p(\alpha)$ è un campo finito di p^d elementi. Per la Proposizione 2.2, α è radice di $x^{p^d} - x$. Inoltre, sempre per la Proposizione 2.2, α non è radice di $x^{p^j} - x$ per nessun $j < d$ (perché le radici in $\mathbb{F}_p(\alpha)$ di tale polinomio formerebbero un sottocampo contenente α e di al più p^j elementi, il che è assurdo). Quindi i coniugati $\alpha, \alpha^p, \alpha^{p^2}, \dots, \alpha^{p^{d-1}}$ sono distinti, mentre $\alpha^{p^d} = \alpha$. (In altre parole, α appartiene ad un'orbita lunga d sotto l'azione di $\langle \sigma \rangle$, il gruppo di Galois di $\mathbb{F}_q/\mathbb{F}_p$. Ne seguirebbe anche subito che $d \mid f$, se $q = p^f$, ma per questo vedi la Proposizione successiva.)

Inoltre, poiché $g(x)$ ha coefficienti in \mathbb{F}_p e $\beta^p = \beta$ per ogni $\beta \in \mathbb{F}_p$, abbiamo $g(\alpha^{p^j}) = (g(\alpha))^{p^j} = 0$, e quindi ogni $\sigma^j(\alpha)$ è radice di $g(x)$. Essendo $\alpha, \alpha^p, \dots, \alpha^{p^{d-1}}$ elementi distinti, questi sono *tutte* le radici di $g(x)$, pertanto $g(x) = \prod_{j=0}^{d-1} (x - \alpha^{p^j})$. \square

Segue dalla Proposizione che ogni polinomio $g(x)$ irriducibile su \mathbb{F}_p di grado d divide $x^{p^d} - x$, un fatto che riprenderemo presto. Notate inoltre che non appena un campo finito \mathbb{F}_q contiene una radice di un polinomio $g(x)$ irriducibile su \mathbb{F}_p , le contiene tutte. (In altre parole, il campo di spezzamento di polinomi irriducibili su un campo finito si costruisce “in un colpo solo”, a differenza ad esempio di quanto avviene sui razionali.)

PROPOSIZIONE 2.4. *Ogni sottocampo di \mathbb{F}_{p^f} ha p^d elementi, con $d \mid f$. Viceversa, se $d \mid f$ esiste esattamente un sottocampo di \mathbb{F}_{p^f} con p^d elementi, che quindi possiamo indicare senza ambiguità con \mathbb{F}_{p^d} .* ²

DIMOSTRAZIONE. Un sottocampo F di \mathbb{F}_{p^f} è un campo con p^d elementi, ma \mathbb{F}_{p^f} è uno spazio vettoriale su F e quindi p^f è una potenza di p^d , cioè d divide f . Inoltre F è l'unico sottocampo di \mathbb{F}_{p^f} del suo ordine, essendo individuato come l'insieme delle radici di $x^{p^d} - x$ in \mathbb{F}_{p^f} . ³

Viceversa se d divide f ogni soluzione di $x^{p^d} = x$ (in qualunque campo) è anche soluzione di $x^{p^f} = x$. Pertanto $x^{p^d} - x$ divide $x^{p^f} - x$, in quanto entrambi hanno radici distinte. ⁴ Quindi $x^{p^d} - x$ si fattorizza completamente in \mathbb{F}_{p^f} , e l'insieme delle sue radici è un sottocampo di p^d elementi. \square

OSSERVAZIONI. In generale se $F \subseteq E$ sono campi, allora F^* è un sottogruppo di E^* . Ma naturalmente non ogni sottogruppo di $\mathbb{F}_{p^f}^*$ è il gruppo moltiplicativo di un sottocampo. Però se ha l'ordine giusto per esserlo, cioè una potenza di p , meno

²Naturalmente la Proposizione segue immediatamente dalla teoria di Galois, dove i sottocampi corrispondono ai sottogruppi del gruppo di Galois $\langle \sigma \rangle$. Vogliamo però darne una dimostrazione indipendente, e non assumiamo la Teoria di Galois.

³Un altro modo, forse un po' più complicato, di mostrare che d divide f è notare che l'automorfismo di Frobenius di \mathbb{F}_{p^f} ha ordine f , mentre la sua restrizione a F , che evidentemente è l'automorfismo di Frobenius di F , ha ordine d ; quindi d divide f .

⁴Alternativamente, usate il fatto che $p^d - 1 \mid p^f - 1$ se $d \mid f$ (vedi la prossima Osservazione), per cui $x^{p^d-1} - 1$ divide $x^{p^f-1} - 1$, (indipendentemente dal fatto che il primo p sia la caratteristica del campo).

1, allora lo è. Ciò segue dalla Proposizione precedente, assieme al fatto che $\mathbb{F}_{p^f}^*$, essendo ciclico, ha esattamente un sottogruppo di ciascun ordine che divide il suo.

Un altro modo di mostrare questo fatto è dedurlo dal fatto che $(p^d - 1) \mid (p^f - 1) \Leftrightarrow d \mid f$, che ora dimostriamo.

Per il verso meno semplice (\Rightarrow), che è quello che ci serve qui, basta notare che se $(p^d - 1) \mid (p^f - 1)$ allora $x^{p^d} - x = x(x^{p^d-1} - 1)$ fattorizza completamente in \mathbb{F}_{p^f} , e quindi l'insieme delle sue radici è un sottocampo di \mathbb{F}_{p^f} di ordine p^d , da cui $d \mid f$ come visto nella Proposizione 2.2. (Il verso opposto è immediato.)

In realtà è facile mostrare che piú in generale $(n^d - 1) \mid (n^f - 1) \Leftrightarrow d \mid f$, qualunque sia $n \in \mathbb{Z}$ con $n \neq 0, \pm 1$ e $(n, d) \neq (-2, 2)$ (dove $-3 \mid (-2)^f - 1$ per ogni f).

Infatti il verso (\Leftarrow) è facile, se $f = ds$ allora $n^f - 1 = (n^d - 1)(n^{d(s-1)} + n^{d(s-2)} + \dots + n^d + 1)$. Viceversa per (\Rightarrow), scriviamo $f = dq + r$ con $0 \leq r < d$, allora $n^f - 1 = (n^{dq} - 1)n^r + n^r - 1 \equiv n^r - 1 \pmod{n^d - 1}$, ma anche $n^f - 1 \equiv 0 \pmod{n^d - 1}$ per ipotesi, ed essendo $|n^r - 1| < n^d - 1$ segue che $r = 0$.

E' interessante notare che mentre $(x^d - 1) \mid (x^f - 1)$ in $\mathbb{Z}[x]$ implica ovviamente $(n^d - 1) \mid (n^f - 1)$ per ogni $n \in \mathbb{Z}$, è sufficiente la validità della seconda per un singolo $n \in \mathbb{Z} \setminus \{0, \pm 1, -2\}$ per implicare la prima.

PROPOSIZIONE 2.5. *Se $q = p^f$, il polinomio $x^q - x$ fattorizza in $\mathbb{F}_p[x]$ nel prodotto di tutti i polinomi monici irriducibili di grado d che divide f .*

DIMOSTRAZIONE. Abbiamo visto che

$$x^q - x = \prod_{\alpha \in \mathbb{F}_q} (x - \alpha).$$

Ogni orbita di \mathbb{F}_q sotto l'azione di $\langle \sigma \rangle$ è del tipo $\alpha, \alpha^p, \alpha^{p^2}, \dots, \alpha^{p^d-1}$, per qualche $\alpha \in \mathbb{F}_q$, dove $d = |\mathbb{F}_p(\alpha) : \mathbb{F}_p|$, e quindi $d \mid f$, come osservato in precedenza. Per quanto visto $\prod_{j=0}^{d-1} (x - \alpha^{p^j})$ è il polinomio minimo di α su \mathbb{F}_p , in particolare è monico

irriducibile in $\mathbb{F}_p[x]$, di grado $d \mid f$. Quindi $x^q - x$ è prodotto di polinomi monici irriducibili in $\mathbb{F}_p[x]$ i cui gradi sono divisori di f . Tali polinomi sono distinti in quanto $x^q - x$ ha radici distinte.

Viceversa se $g(x)$ è un polinomio monico irriducibile su \mathbb{F}_p di grado $d \mid f$, allora $\mathbb{F}_p[x]/(g(x))$ è un campo finito con p^d elementi, e quindi $g(x) \mid (x^{p^d} - x)$ per quanto visto, inoltre $(x^{p^d} - x) \mid (x^q - x)$. \square

COROLLARIO 2.6. *Per ogni intero positivo d indichiamo con n_d il numero di polinomi irriducibili di grado d su \mathbb{F}_p . allora per ogni intero positivo f vale $\sum_{d \mid f} dn_d = p^f$. Questo permette di calcolare induttivamente n_f come*

$$n_f = \frac{1}{f} \left(p^f - \sum_{d \mid f, d < f} dn_d \right)$$

(partendo da $n_1 = p$). In particolare $n_f = (p^f - p)/f$ se f è primo.

DIMOSTRAZIONE. Abbiamo visto che $x^{p^f} - x$ è il prodotto di tutti i polinomi irriducibili su \mathbb{F}_p di grado d che divide f , e quelli di grado d sono in numero di n_d . \square

OSSERVAZIONI. Applicando la formula di inversione di Möbius a $\sum_{d|f} dn_d = p^f$ otteniamo una formula non ricorsiva per n_f :

$$n_f = \frac{1}{f} \sum_{d|f} \mu(d) p^{f/d}.$$

Ad esempio, $n_{12} = \frac{1}{12}(p^{12} - p^6 - p^4 + p^2)$.

Naturalmente la costruzione del campo \mathbb{F}_{p^f} come campo di spezzamento del polinomio $x^{p^f} - x$ su \mathbb{F}_p , vista nella dimostrazione della Proposizione 2.2 è utile dal punto di vista teorico (perché fornisce la dimostrazione dell'esistenza di \mathbb{F}_{p^f}), ma è poco pratica. Invece, un campo con p^f elementi verrà costruito in pratica come $\mathbb{F}_p[x]/F(x)$, dove $F(x)$ è un qualunque polinomio irriducibile su \mathbb{F}_p , di grado f . L'esistenza di almeno un tale polinomio è garantita dalla stessa esistenza del campo \mathbb{F}_{p^f} : il polinomio minimo su \mathbb{F}_p di un generatore del gruppo moltiplicativo di \mathbb{F}_{p^f} soddisfa le richieste. (Comunque, ci saranno in generale anche polinomi irriducibili di grado f le cui radici non sono generatori di $\mathbb{F}_{p^f}^*$: per costruire il campo vanno tanto bene quanto $F(x)$.) Grazie alla Proposizione appena vista, un modo di trovare $F(x)$ è cercarlo fra i divisori irriducibili di $x^{p^f} - x$.

ESEMPIO 2.7. Abbiamo $\mathbb{F}_4 = \mathbb{F}_2(\alpha)$ con $\alpha^2 + \alpha + 1 = 0$ (modo abbreviato di dire “con α una radice del polinomio irriducibile $x^2 + x + 1$ ”), ed anche $\mathbb{F}_{25} = \mathbb{F}_5(\alpha)$ con $\alpha^2 + \alpha + 1 = 0$. In questi due casi la ricerca di un polinomio irriducibile di grado 2 è stata facilitata dal seguente trucco: sia \mathbb{F}_4 che \mathbb{F}_{25} contengono un elemento α di ordine (moltiplicativo) 3, che invece non appartiene a nessun sottocampo proprio (in questo caso, al loro campo primo); dunque si ottengono come $\mathbb{F}(\alpha)$, dove α è una radice del polinomio $x^3 - 1$ che non stia nel campo primo; essendo $x^3 - 1 = (x - 1)(x^2 + x + 1)$, α dovrà essere radice di $(x^2 + x + 1)$, e questo è necessariamente irriducibile sul campo primo (senza bisogno di verifiche).

Analogamente ottengo $\mathbb{F}_9 = \mathbb{F}_3(\alpha)$ con $\alpha^2 + 1 = 0$ (notando che $x^4 - 1 = (x - 1)(x + 1)(x^2 + 1)$), oppure $\mathbb{F}_{16} = \mathbb{F}_2(\alpha)$ con $\alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1 = 0$ e $\mathbb{F}_{81} = \mathbb{F}_3(\alpha)$ con $\alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1 = 0$ (notando che $x^5 - 1 = (x - 1)(x^4 + x^3 + x^2 + x + 1)$, e che $x^4 + x^3 + x^2 + x + 1$ deve essere per forza irriducibile sul campo primo perché ha il grado giusto ed una sua radice, avendo ordine 5, non può appartenere ad alcun sottocampo proprio).

Naturalmente il trucco non funziona in altri casi, quindi ad esempio $\mathbb{F}_8 = \mathbb{F}_2(\alpha)$ con $\alpha^3 + \alpha + 1$, dove il polinomio $x^3 + x + 1$, irriducibile di grado 3, è stato determinato per prove ed errori (a meno di non saper “indovinare” la fattorizzazione $x^7 - 1 = (x^3 + x + 1)(x^3 + x^2 + 1)$ su \mathbb{F}_2). Oppure, per fare i calcoli in \mathbb{F}_{16} la presentazione $\mathbb{F}_{16} = \mathbb{F}_2(\alpha)$ con $\alpha^4 + \alpha + 1$ potrebbe essere piú conveniente di quella trovata in precedenza (ma bisogna aver verificato che $x^4 + x + 1$ è irriducibile su \mathbb{F}_2).

PROPOSIZIONE 2.8. *Sia \mathbb{F}_q un campo finito, con $q = p^f$. Allora due elementi di \mathbb{F}_q si possono moltiplicare o dividere in $O(\log^2 q)$ operazioni bit (esattamente la stessa stima che per moltiplicare o dividere (in quel caso con resto) due elementi di $\mathbb{Z}/q\mathbb{Z}$).*

Se k è un intero positivo, allora un elemento di \mathbb{F}_q si può elevare alla k -esima potenza in $O(\log k \log^2 q)$ operazioni bit.

DIMOSTRAZIONE. Sia $F(x)$ un polinomio irriducibile di grado f su \mathbb{F}_p . Allora $\mathbb{F}_q \cong \mathbb{F}_p[x]/F(x)$, e quindi ogni elemento di \mathbb{F}_q si rappresenta in modo unico come polinomio di grado minore di f . Ora il punto è che polinomi a coefficienti in \mathbb{F}_p si addizionano e moltiplicano in modo simile ai numeri interi scritti in base p , salvo l'assenza di riporti nel caso dei polinomi.

Per moltiplicare due elementi di \mathbb{F}_q bisogna eseguire al più f^2 moltiplicazioni di interi modulo p (e un certo numero di addizioni, che comunque richiedono meno tempo), ed infine ridurre il prodotto modulo $F(x)$. Quest'ultima divisione fra polinomi richiede $O(f)$ divisioni di interi modulo p e $O(f^2)$ moltiplicazioni di interi modulo p . Ciascuna moltiplicazione di interi modulo p richiede $O(\log^2 p)$ operazioni bit, ed anche ciascuna divisione di interi modulo p richiede $O(\log^2 p)$ operazioni bit, perché si fa con l'algoritmo di Euclide, per il quale abbiamo visto la stima migliorata $O(\log^2 p)$ rispetto a $O(\log^3 p)$. In definitiva la moltiplicazione di due elementi di \mathbb{F}_q richiede $O(f^2 \log^2 p) = O(\log^2 q)$ operazioni bit.

Per la divisione in \mathbb{F}_q basta saper calcolare l'inverso di un polinomio $g(x)$ modulo $F(x)$, e questo (come per calcolare l'inverso di un intero n modulo un intero coprimo m) si fa con l'algoritmo di Euclide per i polinomi e richiede $O(\log^2 q)$ operazioni bit (la stessa stima, anche qui, che per interi della stessa *grandezza*, perché ciascuna addizione, moltiplicazione ecc. di polinomi richiede la stessa stima in operazioni bit che interi della stessa *grandezza*).

Infine una k -esima potenza modulo $F(x)$ si calcola con il metodo dei quadrati ripetuti e (come la k -esima potenza di un intero modulo q) richiede $O(\log k \log^2 q)$ operazioni bit. \square

2.2. Resti quadratici e reciprocità

Sappiamo come risolvere le congruenze lineari (cioè di primo grado, ed anche i sistemi di congruenze lineari). Vorremmo vedere come si risolvono quelle di secondo grado

$$ax^2 + bx + c \equiv 0 \pmod{m}$$

Anzitutto il Teorema Cinese dei resti permette di ridursi a congruenze del tipo

$$ax^2 + bx + c \equiv 0 \pmod{p^\alpha}, \quad \text{con } p \text{ primo}$$

Il primo passo per risolvere quest'ultima è risolvere

$$ax^2 + bx + c \equiv 0 \pmod{p}$$

Possiamo assumere che p non divida a , altrimenti la congruenza è lineare. Se $p = 2$ è facile: la congruenza ha la forma $x^2 + c \equiv 0 \pmod{2}$, che ha l'unica soluzione (unica modulo 2) $x \equiv 0$ se $c \equiv 0$, o $x \equiv 1$ se $c \equiv 1$, oppure la forma $x^2 + x + c \equiv 0 \pmod{2}$, che ha due o nessuna soluzione a seconda che $c \equiv 0$ o $c \equiv 1$ modulo 2.

Supponiamo che p sia dispari. Moltiplichiamo la congruenza per $4a$ e scriviamola nella forma

$$(2ax + b)^2 \equiv b^2 - 4ac \pmod{p}$$

Se poniamo $y = 2ax + b$ e $\Delta = b^2 - 4ac$, il problema diventa quello di risolvere $y^2 \equiv \Delta \pmod{p}$, cioè di estrarre radici quadrate modulo p . Una volta risolto questo, ciascuna soluzione per y darà luogo ad una soluzione per x tramite $x = (-b + y)/(2a)$.

OSSERVAZIONI. Lo stesso metodo risolverà anche la congruenza modulo p^α nel caso p dispari e p che non divide a (estraendo radici quadrate modulo p^α), solo che questo non sarà più il caso generale (perché rimane escluso il caso in cui a è multiplo di p ma non di p^α).

2.2.1. Radici dell'unità.

PROPOSIZIONE 2.9. *Se n è un intero positivo, il campo \mathbb{F}_q contiene esattamente $(n, q-1)$ radici n -esime dell'unità, cioè soluzioni dell'equazione $x^n = 1$. In particolare \mathbb{F}_q contiene una radice n -esima primitiva dell'unità ξ (cioè un elemento di ordine moltiplicativo esattamente n) se e solo se n divide $q-1$. Se ξ è una radice n -esima primitiva dell'unità in \mathbb{F}_q , allora ξ^j è radice n -esima primitiva sse $(j, n) = 1$.*

DIMOSTRAZIONE. Una radice n -esima dell'unità in \mathbb{F}_q è un elemento del gruppo ciclico \mathbb{F}_{q-1}^* di ordine un divisore di n (ed ovviamente anche di $q-1$). Tali elementi formano l'unico sottogruppo (ciclico) di \mathbb{F}_{q-1}^* di ordine $(n, q-1)$. Il resto è noto. \square

COROLLARIO 2.10. *L'elemento -1 di \mathbb{F}_q con q dispari ha una radice quadrata in \mathbb{F}_q sse $q \equiv 1 \pmod{4}$ (perché una tale radice quadrata è una radice quarta primitiva dell'unità).*

Grazie al Corollario, se $q \equiv 3 \pmod{4}$ il polinomio $x^2 + 1$ è irriducibile su \mathbb{F}_q , e quindi $\mathbb{F}_q[x]/(x^2 + 1) \cong \mathbb{F}_{q^2}$. In altre parole, $\mathbb{F}_{q^2} = \mathbb{F}_q(i)$, dove i è una radice quarta primitiva dell'unità (come i in \mathbb{C}).

In particolare se $q = p$ è primo, sempre con $p \equiv 3 \pmod{4}$, possiamo realizzare \mathbb{F}_{p^2} come quoziente dell'anello $\mathbb{Z}[i] = \{a + bi : a, b \in \mathbb{Z}\}$ degli interi di Gauss modulo l'ideale $p\mathbb{Z}[i]$ (cioè riguardare i coefficienti modulo p). Infatti tale quoziente ha p^2 elementi, è un dominio e quindi un campo perché ogni dominio finito è un campo. [Infatti, dire che $\mathbb{Z}[i]/p\mathbb{Z}[i]$ è un dominio equivale a dire che p è un primo (o equivalentemente un irriducibile, essendo $\mathbb{Z}[i]$ un PID) in $\mathbb{Z}[i]$ (oltre che in \mathbb{Z}). Per mostrare che p è primo in $\mathbb{Z}[i]$ basta notare che non può essere $p = (a + ib)(a - ib) = a^2 + b^2$, perché una somma di quadrati è $\equiv 0, 1, 2 \pmod{4}$. Notate che se invece $p \equiv 1 \pmod{4}$, allora $\mathbb{Z}[i]/p\mathbb{Z}[i]$ non è un dominio, cioè p non è primo in $\mathbb{Z}[i]$, in quanto si può dimostrare che ogni primo $\equiv 1 \pmod{4}$ è somma di due quadrati, cioè $p = a^2 + b^2 = (a + ib)(a - ib)$ con a, b opportuni interi positivi.]

2.2.2. Resti quadratici. Se p è un primo dispari e $a \in \mathbb{Z}$ con $(a, p) = 1$, diciamo che a è un *resto quadratico modulo p* se la congruenza $x^2 \equiv a \pmod{p}$ ha soluzioni (ovvero se l'equazione $x^2 = a$ ha soluzioni in \mathbb{F}_p , pensando $a \in \mathbb{F}_p$; se ne ha, ne ha esattamente due, opposte fra loro); diciamo che a è un *non-resto quadratico modulo p* altrimenti. Ciò dipende solo dal resto di a modulo p .

Definiamo il *simbolo di Legendre* come segue

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & \text{se } p/a \\ 1 & \text{se } a \text{ è un resto quadratico modulo } p \\ -1 & \text{se } a \text{ è un non-resto quadratico modulo } p \end{cases}$$

Possiamo notare che in ogni caso $1 + \left(\frac{a}{p}\right)$ è il numero di soluzioni in \mathbb{F}_p dell'equazione $x^2 = a$

PROPOSIZIONE 2.11 (Eulero).

$$\left(\frac{a}{p}\right) \equiv a^{\frac{p-1}{2}} \pmod{p}$$

Notiamo che applicare questa formula ci da una stima $O(\log^3 p)$ per decidere se a è un resto quadratico modulo p .

DIMOSTRAZIONE. La mappa

$$\begin{array}{ccc} \mathbb{F}_p^* & \rightarrow & \mathbb{F}_p^* \\ x & \mapsto & x^2 \end{array}$$

è un endomorfismo del gruppo ciclico \mathbb{F}_p^* , che ha ordine pari, quindi il suo nucleo ha ordine 2 e quindi la sua immagine è l'unico sottogruppo di \mathbb{F}_p^* di ordine $(p-1)/2$, cioè l'insieme delle radici $(p-1)/2$ -esime dell'unità in \mathbb{F}_p^* . Questo dimostra il caso $(a, p) = 1$, e il caso opposto è ovvio. \square

COROLLARIO 2.12. (a) $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right) \left(\frac{b}{p}\right)$

(b) se p non divide b , allora $\left(\frac{ab^2}{p}\right) = \left(\frac{a}{p}\right)$

(c) $\left(\frac{1}{p}\right) = 1$ e $\left(\frac{-1}{p}\right) = (-1)^{(p-1)/2}$

Il corollario permette di ridurre il calcolo di $\left(\frac{a}{p}\right)$ al calcolo di $\left(\frac{q}{p}\right)$ per ogni fattore primo q di a (qui q è un altro primo, non più una potenza di p), purché sappiamo fattorizzare a in un prodotto di primi.

TEOREMA 2.13. ⁵

$$\left(\frac{2}{p}\right) = (-1)^{(p^2-1)/8} = \begin{cases} 1 & \text{se } p \equiv \pm 1 \pmod{8} \\ -1 & \text{se } p \equiv \pm 3 \pmod{8} \end{cases}$$

⁵Notate che l'esponente $(p^2-1)/8$ nella formula è intero, e che il suo valore dipende solo dal resto di p modulo 8. La prima affermazione segue da $(2a+1)^2 = 4a(a+1) + 1 \equiv 1 \pmod{8}$; per la seconda, se $p = 8b + c$ (quindi c è dispari) allora $(p^2-1)/8 = 2b(4b+c) + (c^2-1)/8$.

TEOREMA 2.14 (Legge di Reciprocità Quadratica di Gauss). *Se p, q sono primi dispari (distinti) vale*

$$\left(\frac{q}{p}\right) = (-1)^{(p-1)(q-1)/4} \left(\frac{p}{q}\right) = \begin{cases} -\left(\frac{p}{q}\right) & \text{se } p \equiv q \equiv 3 \pmod{4} \\ \left(\frac{p}{q}\right) & \text{altrimenti} \end{cases}$$

La legge di Reciprocità Quadratica e le due leggi accessorie su $\left(\frac{-1}{p}\right)$ e $\left(\frac{2}{p}\right)$ permettono di calcolare induttivamente qualsiasi simbolo di Legendre $\left(\frac{a}{p}\right)$. Infatti si può assumere $0 \leq a < p$ (o volendo anche $a < p/2$) riducendo $a \pmod{p}$ e poi, fattorizzando a , ridursi al calcolo di certi $\left(\frac{q_i}{p}\right)$ con primi dispari q_i, p , e quindi, grazie alla Legge di Reciprocità Quadratica, ai corrispondenti $\left(\frac{p}{q_i}\right)$, che ora si possono ridurre a loro volta, ecc.

ESEMPIO 2.15.

$$\begin{aligned} \left(\frac{-42}{61}\right) &= \left(\frac{-1}{61}\right) \left(\frac{2}{61}\right) \left(\frac{3}{61}\right) \left(\frac{7}{61}\right) = 1 \cdot (-1) \left(\frac{61}{3}\right) \left(\frac{61}{7}\right) \\ &= -\left(\frac{1}{3}\right) \left(\frac{5}{7}\right) = -\left(\frac{7}{5}\right) = -\left(\frac{2}{5}\right) = 1 \end{aligned}$$

Un modo più veloce era

$$\left(\frac{-42}{61}\right) = \left(\frac{19}{61}\right) = \left(\frac{61}{19}\right) = \left(\frac{4}{19}\right) = 1$$

Dunque -42 (o 19) ha almeno una radice quadrata modulo 61 , e quindi ne ha esattamente due (poiché se ne ha almeno una allora si trovano tutte moltiplicandola per le radici quadrate dell'unità, in \mathbb{F}_{61} , vale a dire ± 1).⁶

ESEMPIO 2.16. Vogliamo determinare tutti i primi $p > 3$ tali che la congruenza $x^2 \equiv 3 \pmod{p}$ abbia soluzioni. Grazie alla LRQ abbiamo

$$\left(\frac{3}{p}\right) = (-1)^{(p-1)/2} \left(\frac{p}{3}\right).$$

Inoltre $(-1)^{(p-1)/2} \equiv p \pmod{4}$ e $\left(\frac{p}{3}\right) \equiv p \pmod{3}$. Concludiamo che $\left(\frac{3}{p}\right) = 1$, cioè $x^2 \equiv 3 \pmod{p}$ ha soluzioni (con $p > 3$) sse $p \equiv \pm 1 \pmod{12}$.

ESEMPIO 2.17. Nello stesso modo si può decidere per quali primi una certa congruenza quadratica ha soluzione. Ad esempio l'equazione $x^2 + x + 1 \equiv 0 \pmod{p}$ è equivalente a $(x + \frac{1}{2})^2 \equiv -\frac{3}{4} \pmod{p}$, che ha soluzioni sse $-\frac{3}{4}$, o,

⁶In effetti, si può verificare che $(\pm 18)^2 \equiv 19 \pmod{61}$; vedremo presto come calcolare tali radici quadrate.

equivalentemente, -3 , è un quadrato modulo p . Ora

$$\begin{aligned} \left(\frac{-3}{p}\right) &= \left(\frac{-1}{p}\right) \left(\frac{3}{p}\right) = (-1)^{(p-1)/2} (-1)^{(p-1)/2} \left(\frac{p}{3}\right) \\ &= \begin{cases} 1 & \text{se } p \equiv 1 \pmod{3} \\ -1 & \text{se } p \equiv -1 \pmod{3} \end{cases} \end{aligned}$$

Naturalmente la congruenza quadratica che abbiamo scelto si presta anche ad una soluzione piú diretta: una soluzione della congruenza è esattamente una radice cubica primitiva modulo p (se $p > 3$). In questo modo possiamo anche rispondere alla domanda piú generale con un intero m al posto del primo p : i possibili valori di m sono il numero 3, e tutti quelli che sono prodotti di primi $\equiv 1 \pmod{3}$.

ESERCIZIO 2.18. Svolgere l'esercizio analogo con $x^2 - x + 1 \equiv 0$.

ESEMPIO 2.19. Se $F_n = 2^{2^n} + 1$ è primo, cioè un primo di Fermat, allora 5 è una radice primitiva modulo F_n , ad eccezione del caso di $F_1 = 5$. Per vederlo basta verificare che 5 non sia un resto quadratico modulo F_n . Questo è chiaramente vero per $n = 0$, mentre per $n \geq 2$ basta calcolare

$$\left(\frac{5}{F_n}\right) = \left(\frac{5}{2^{2^n} + 1}\right) = \left(\frac{2^{2^n} + 1}{5}\right) = \left(\frac{1 + 1}{5}\right) = -1,$$

dove abbiamo usato il fatto che $2^4 \equiv 1 \pmod{5}$.

ESERCIZIO 2.20. Mostrate in modo analogo che 3 e 7 sono radici primitive modulo ogni primo di Fermat diverso da $F_0 = 3$.

ESEMPIO 2.21 (Teorema di Pépin). Il numero di Fermat $F_n = 2^{2^n} + 1$, con $n \geq 1$, è primo se e solo se $3^{2^{2^n-1}} \equiv -1 \pmod{F_n}$. Infatti se F_n è primo allora 3 è un non-resto quadratico modulo F_n grazie all'esercizio precedente, e quindi $3^{(F_n-1)/2} \equiv -1 \pmod{F_n}$ grazie alla Proposizione di Eulero 2.11. Viceversa, se vale quest'ultima condizione allora 3 ha ordine esattamente $2^{2^n} = F_n - 1$ modulo qualsiasi divisore primo p di F_n , e quindi $p = F_n$.

Ad esempio, il seguente calcolo *dimostra* (anche se in modo inefficiente per un numero così piccolo) che $F_3 = 2^8 + 1 = 257$ è primo, usando ripetutamente il fatto che $2^8 \equiv -1$:

$$\begin{aligned} 3^2 &\equiv 9 = 2^3 + 1 \pmod{F_3} \\ 3^4 &\equiv 2^6 + 2^4 + 1 \\ 3^8 &\equiv 2^{12} + 2^8 + 1 + 2 \cdot 2^{10} + 2 \cdot 2^6 + 2 \cdot 2^4 \equiv -2^4 - 2^3 + 2^7 + 2^5 = 2^7 + 2^3 \\ 3^{16} &\equiv 2^{14} + 2^6 + 2 \cdot 2^{10} \equiv -2^3 \\ 3^{32} &\equiv 2^6 \\ 3^{64} &\equiv 2^{12} \equiv -2^4 \\ 3^{128} &\equiv 2^8 \equiv -1 \end{aligned}$$

ESERCIZIO 2.22. Verificate in modo analogo che $F_4 = 2^{16} + 1 = 65537$ è primo.

ESERCIZIO 2.23. Se $M_n = 2^p - 1$ è primo, cioè un primo di Mersenne, allora 3 è un non-resto quadratico modulo M_p , ad eccezione del caso di $M_2 = 3$.

DIMOSTRAZIONE DEL TEOREMA 2.13. Esiste un'unica funzione $f(n)$ di n intero positivo tale che $f(p) = (-1)^{(p^2-1)/8}$ per p primo dispari, $f(2) = 0$, e che sia completamente moltiplicativa, cioè che $f(mn) = f(m)f(n)$ per ogni m, n , ed è $f(n) = (-1)^{(n^2-1)/8}$ per n dispari, $f(n) = 0$ per n pari. Che $f(n)$ sia moltiplicativa si vede dal fatto che

$$\begin{aligned} \frac{m^2n^2 - 1}{8} &= \frac{m^2 - 1}{8} + \frac{n^2 - 1}{8} + \frac{(m^2 - 1)(n^2 - 1)}{8} \\ &\equiv \frac{m^2 - 1}{8} + \frac{n^2 - 1}{8} \end{aligned}$$

modulo 2, anzi, modulo 8, poiché 8 divide sia $m^2 - 1$ che $n^2 - 1$ per m, n dispari.

Mostreremo che $\left(\frac{2}{p}\right) = f(p)$. Sia ξ una radice ottava primitiva dell'unità in un'estensione di \mathbb{F}_p . Sicuramente si trova in \mathbb{F}_{p^2} , essendo $p^2 \equiv 1 \pmod{8}$. Dunque ξ è una radice di $x^4 + 1$, perciò $\xi^4 = -1$. Definiamo la *somma di Gauss*

$$G = \sum_{j=0}^7 f(j)\xi^j$$

Quindi $G = \xi - \xi^3 - \xi^5 + \xi^7 = 2(\xi - \xi^3)$, e $G^2 = 4(\xi^2 - 2\xi^4 + \xi^6) = 8$ (notare che se fosse $\xi = e^{2\pi i/8}$ in \mathbb{C} , cioè $\xi = (i+1)/\sqrt{2}$, avremmo $G = 2\sqrt{2}$). Ora calcoliamo G^p in due modi diversi:

$$\begin{aligned} G^p &= (G^2)^{(p-1)/2}G = 8^{(p-1)/2}G = \left(\frac{8}{p}\right)G = \left(\frac{2}{p}\right)G \\ G^p &= \sum_{j \in \mathbb{Z}/8\mathbb{Z}} f(j)\xi^{p \cdot j} = f(p) \sum_{j \in \mathbb{Z}/8\mathbb{Z}} f(p \cdot j)\xi^{p \cdot j} = f(p)G \end{aligned}$$

notando che $f(j)^p = f(j)$, che $f(j) = f(p)^2 f(j) = f(p)f(p \cdot j)$, e che $j \mapsto p \cdot j$ è una biiezione di $\mathbb{Z}/8\mathbb{Z}$ su se stesso. Concludiamo confrontando le due uguaglianze per G^p e semplificando G (che non è nullo in \mathbb{F}_{p^2} essendo $G^2 = 8$). \square

DIMOSTRAZIONE DEL TEOREMA 2.14. Sia ξ una radice q -esima primitiva dell'unità in un'estensione di \mathbb{F}_p . La più piccola tale estensione è $\mathbb{F}_p(\xi) = \mathbb{F}_{p^f}$, dove f è l'ordine di p modulo q , cioè l'ordine di p nel gruppo moltiplicativo di \mathbb{F}_q . (Notare lo scambio di ruoli fra p e q , qui è q la caratteristica.) In altre parole, f è il più piccolo intero per cui q divide $p^f - 1$. Sicuramente $\xi \in \mathbb{F}_{p^{q-1}}$, poiché $q \mid (p^{q-1} - 1)$ grazie a Eulero-Fermat. Definiamo la somma di Gauss

$$G = \sum_{j=0}^{q-1} \left(\frac{j}{q}\right) \xi^j.$$

Mostreremo presto che $G^2 = (-1)^{(q-1)/2}q$. Usando questo fatto abbiamo

$$G^p = (G^2)^{(p-1)/2}G = ((-1)^{(q-1)/2}q)^{(p-1)/2}G = (-1)^{(p-1)(q-1)/4} \left(\frac{q}{p}\right)G$$

D'altra parte essendo $\left(\frac{j}{p}\right)^p = \left(\frac{j}{q}\right)$ e $\left(\frac{j}{q}\right)^p = \left(\frac{p}{q}\right)\left(\frac{pj}{q}\right)$ abbiamo

$$G^p = \sum_{j \in \mathbb{Z}/q\mathbb{Z}} \left(\frac{j}{q}\right) \xi^{pj} = \left(\frac{p}{q}\right) \sum_{j \in \mathbb{Z}/q\mathbb{Z}} \left(\frac{pj}{q}\right) \xi^{pj} = \left(\frac{p}{q}\right) G$$

Semplificando $G \neq 0$ (poiché $G^2 = \pm q$) dal confronto delle due uguaglianze otteniamo la conclusione. \square

LEMMA 2.24. Vale $G^2 = (-1)^{(q-1)/2}q$.

DIMOSTRAZIONE.

$$\begin{aligned} G^2 &= \sum_{j,k=1}^{q-1} \left(\frac{j}{q}\right) \xi^j \left(\frac{-k}{q}\right) \xi^{-k} \\ &= \left(\frac{-1}{q}\right) \sum_{j=1}^{q-1} \sum_{k=1}^{q-1} \left(\frac{jk}{q}\right) \xi^{j-k} \\ &= (-1)^{(q-1)/2} \sum_{j=1}^{q-1} \sum_{k=1}^{q-1} \left(\frac{j^2k}{q}\right) \xi^{j(1-k)}. \end{aligned}$$

Qui abbiamo fatto il cambio di variabili $k \mapsto jk$. Riordinando, la doppia sommatoria diventa

$$\sum_{k=1}^{q-1} \left(\frac{k}{q}\right) \sum_{j=0}^{q-1} \xi^{j(1-k)}.$$

Notate che abbiamo incluso anche il termine $j = 0$, dato che fornisce un contributo

$$\sum_{k=1}^{q-1} \left(\frac{k}{q}\right) = 0,$$

dato che ci sono tanti quadrati quanti non-quadrati.

Ora se $k \neq 1$, abbiamo $\sum_{j=0}^{q-1} \xi^{j(1-k)} = 0$, dato che si tratta della somma delle radici $(q-1)$ -esime di 1, che è l'opposto del coefficiente x^{q-2} in $x^{q-1} - 1$, cioè zero.

Otteniamo

$$G^2 = (-1)^{(q-1)/2} \left(\frac{1}{q}\right) \sum_{j=0}^{q-1} 1 = (-1)^{(q-1)/2}q,$$

che conclude la dimostrazione. \square

Nella prossima sezione vedremo una dimostrazione del Lemma leggermente diversa, e forse un po' più elegante.

2.2.3. Somme di Gauss in \mathbb{C} . Sia n un intero maggiore di 1. Il numero complesso $\xi = e^{2\pi i/n}$ è una radice n -esima primitiva di 1 in \mathbb{C} (il più piccolo sottocampo $\mathbb{Q}(\xi)$ di \mathbb{C} che lo contiene è detto un *campo ciclotomico*). È chiaro che ξ è radice del polinomio

$$\frac{x^n - 1}{x - 1} = x^{n-1} + x^{n-2} + \cdots + x + 1$$

Non è difficile mostrare che se n è primo questo polinomio è irriducibile su \mathbb{Q} , e quindi è il polinomio minimo di ξ su \mathbb{Q} ; se n non è primo il polinomio minimo è invece un suo divisore proprio. In ogni caso $\sum_{j=0}^{n-1} \xi^j = 0$, come abbiamo appena visto: geometricamente corrisponde al fatto che le ξ^j sono i vertici di un poligono regolare di n lati ed il suo (bari-) centro è l'origine.

Poniamo ora

$$\begin{cases} R = \sum \{ \xi^j : 0 < j < n, j \text{ è resto quadratico modulo } n \}, \\ N = \sum \{ \xi^j : 0 < j < n, j \text{ è non-resto quadratico modulo } n \}, \end{cases}$$

Dunque $R + N + 1 = 0$. [Attenzione, nella definizione di j resto o non-resto quadratico modulo n (che finora conoscevamo solo per n primo), non richiediamo che $(j, n) = 1$.]

Definiamo la *somma (quadratica) di Gauss*

$$G = \sum_{r=0}^{n-1} \xi^{r^2} = 1 + 2R$$

Ora assumiamo che n sia un primo dispari q .⁷ Tornando alle somme di Gauss, avremo anche

$$G = 1 + 2R = R - N = \sum_{j=0}^{n-1} \left(\frac{j}{q} \right) \xi^j$$

Qui ci starebbe bene un disegno con i numeri ξ^{r^2} nel piano complesso: perché non ce lo mettete voi, magari per $q = 7$ o 11 ?

Le potenze ξ^{r^2} hanno la proprietà di avere direzioni in un certo senso assimilabili a direzioni casuali (e questo ha una varietà di applicazioni pratiche, dalle comunicazioni all'acustica delle sale da concerto!). Se diamo per buono questo fatto ci possiamo aspettare che $|G|$ sia dell'ordine di \sqrt{q} , la distanza media dall'origine percorsa dopo q passi di una passeggiata casuale. In effetti $|G| = \sqrt{q}$, e lo

⁷Notiamo di passaggio che in questo caso R ed N sono le somme delle due orbite delle radici dell'equazione $x^{q-1} + x^{q-2} + \dots + x + 1 = 0$ sotto l'azione dell'unico sottogruppo di indice 2 e del suo gruppo di Galois $\text{Gal}(\mathbb{Q}(\xi)/\mathbb{Q})$, il quale è ciclico di ordine $q-1$. In altre parole $\mathbb{Q}(R) = \mathbb{Q}(N)$ è una estensione quadratica di \mathbb{Q} , e quindi R ed N sono soluzioni di equazioni di secondo grado a coefficienti in \mathbb{Q} , come vedremo esplicitamente. Trovare R ed N è il primo passo da fare volendo esprimere ξ mediante successivi radicali di indici più piccoli possibile (naturalmente $\xi = \sqrt[q]{1}$!), ed è il primo passo per dividere la circonferenza in q parti uguali mediante riga e compasso nel caso in cui $q-1$ è una potenza di 2, cioè in cui q è un primo di Fermat 3, 5, 17, 257, 65537 (i soli noti finora, e forse gli unici).

dimostriamo:

$$\begin{aligned} |G| &= G \cdot \bar{G} = \sum_{r=0}^{q-1} \xi^{r^2} \sum_{s=0}^{q-1} \xi^{-s^2} \quad (\text{poiché } \bar{\xi} = \xi^{-1}) \\ &= \sum_{r,s \in \mathbb{F}_q} \xi^{r^2-s^2} = \sum_{r,s \in \mathbb{F}_q} \xi^{(r-s)(r+s)} = \sum_{j,k \in \mathbb{F}_q} \xi^{jk} = q, \end{aligned}$$

per l'argomento già visto nella dimostrazione del Lemma 2.24. Qui abbiamo tenuto conto che $\bar{\xi} = \xi^{-1}$ e abbiamo effettuato il cambio di indici

$$\begin{cases} j = r + s \\ k = r - s \end{cases} \quad \text{con cambio inverso} \quad \begin{cases} r = \frac{j+k}{2} \\ s = \frac{j-k}{2} \end{cases}$$

essendo q dispari).

Dunque $G = \pm\sqrt{q}$ o $G = \pm i\sqrt{q}$ nei due casi. ⁸

Determinare il segno esatto è più difficile (questo non ha senso in un campo finito, ha senso solo in \mathbb{C} e avendo fissato $\xi = e^{2\pi i/q}$, non una qualsiasi radice q -esima dell'unità), ma si trova che è quello positivo in entrambi i casi. Anzi vale il risultato più generale seguente, per ogni $n > 0$:

$$\sum_{r=0}^{n-1} (e^{2\pi i/n})^{r^2} = \begin{cases} (1+i)\sqrt{n} & \text{se } n \equiv 0 \pmod{4} \\ \sqrt{n} & \text{se } n \equiv 1 \pmod{4} \\ 0 & \text{se } n \equiv 2 \pmod{4} \\ i\sqrt{n} & \text{se } n \equiv 3 \pmod{4} \end{cases}$$

2.2.4. L'inizio di un'altra dimostrazione della LRQ. Sono note più di cento dimostrazioni diverse della Legge di Reciprocità Quadratica. Il solo Gauss ne diede una mezza dozzina. Una di queste, fra le più elementari, si basa sul seguente Lemma.

LEMMA 2.25 (Lemma di Gauss). *Siano p un primo dispari ed $a \not\equiv 0 \pmod{p}$. Consideriamo i resti modulo p minimi in valore assoluto degli interi $a, 2a, \dots, \frac{p-1}{2}a$. Se μ è il numero di quelli negativi fra tali resti, allora $\left(\frac{a}{p}\right) = (-1)^\mu$.*

Per resti modulo p minimi in valore assoluto si intendono i resti compresi fra $-p/2$ e $p/2$. Evidentemente essi sono tutti non nulli, se $a \not\equiv 0 \pmod{p}$. Equivalentemente, si possono considerare i resti minimi positivi, ed indicare con μ il numero di quelli maggiori di $p/2$.

DIMOSTRAZIONE. Siano r_1, \dots, r_λ i resti positivi e $-s_1, \dots, -s_\mu$ quelli negativi. È chiaro che gli r_i presi assieme ai $-s_j$ sono tutti distinti (modulo p). Ma anche gli r_i assieme agli s_j sono tutti distinti. Infatti, se fosse $r_i \equiv s_j \pmod{p}$ per qualche i, j , allora scrivendo $r_i = \rho a$ e $-s_j = \sigma a$ per opportuni $0 < \rho, \sigma \leq (p-1)/2$

⁸Che G sia reale nel primo caso, in cui $q \equiv 1 \pmod{4}$, si poteva vedere indipendentemente dal calcolo esplicito appena fatto notando che in questo caso l'opposto di ciascun resto quadratico è un resto quadratico, perciò assieme a ciascun addendo della somma di Gauss compare anche il suo coniugato.

avremo che $(\rho + \sigma)a \equiv 0 \pmod{p}$, da cui $p \mid \rho + \sigma$, il che è impossibile. Quindi gli r_i assieme agli s_j sono i numeri $1, 2, \dots, \frac{p-1}{2}$ in qualche ordine. Dunque

$$\begin{aligned} a(2a) \cdots \left(\frac{p-1}{2}a\right) &\equiv r_1 \cdots r_\lambda (-s_1) \cdots (-s_\mu) \pmod{p} \\ &= (-1)^\mu r_1 \cdots r_\lambda s_1 \cdots s_\mu = (-1)^\mu 1 \cdot 2 \cdots \frac{p-1}{2}. \end{aligned}$$

D'altra parte, $a(2a) \cdots \left(\frac{p-1}{2}a\right) = a^{\frac{p-1}{2}} \cdot 1 \cdot 2 \cdots \frac{p-1}{2}$. □

Ora possiamo usare il Lemma di Gauss per dare un'altra dimostrazione del fatto che $\left(\frac{2}{p}\right) = (-1)^{(p^2-1)/8}$ se p è un primo dispari. Usare il lemma per dimostrare la Legge di Reciprocità Quadratica di Gauss è possibile, ma piú difficile, e non lo facciamo qui (si veda ad esempio [NZ72, Chapter 3]).

SECONDA DIMOSTRAZIONE DEL TEOREMA 2.13. Calcoliamo $\left(\frac{2}{p}\right)$ applicando il Lemma di Gauss con $a = 2$. Degli interi $2, 4, \dots, p-1$, quelli strettamente minori di $p/2$ sono $2, 4, \dots, 2 \lfloor p/4 \rfloor$, quindi $\mu = (p-1)/2 - \lfloor p/4 \rfloor$, che è pari se $p \equiv \pm 1 \pmod{8}$ e dispari se $p \equiv \pm 3 \pmod{8}$, esattamente come l'intero $(p^2-1)/8$. Questa affermazione sulla parità di μ è verificabile direttamente, ma si può anche provarla in modo piú concettuale notando che per ogni intero n dispari vale $\lfloor n/4 \rfloor \equiv (n-1)(n-3)/8 \pmod{2}$. Infatti fra i due numeri $(n-3)/4$ ed $(n-1)/4$ esattamente uno è intero, ed è $\lfloor n/4 \rfloor$, mentre il doppio dell'altro è dispari, cioè $\equiv 1 \pmod{2}$. □

ESERCIZIO 2.26. Usando la stessa idea ed il Teorema di Wilson, mostrate che per ogni primo q ed intero n vale $\lfloor n/q \rfloor \equiv -n(n-1) \cdots (n-q+1)/q \pmod{q}$.

2.2.5. Il simbolo di Jacobi. Quanto visto finora ci permette di calcolare $\left(\frac{a}{p}\right)$ usando la Legge di Reciprocità mediante riduzioni modulo un primo, che si eseguono rapidamente (in tempo polinomiale), e fattorizzazioni di interi $\leq |a|$, che invece non si eseguono in maniera efficiente. Queste ultime si possono evitare introducendo il simbolo di Jacobi, che generalizza il simbolo di Legendre ammettendo al denominatore qualsiasi intero positivo dispari al posto di p . L'idea è quella di estendere il simbolo di Legendre (che è completamente moltiplicativo nel "numeratore") ad un simbolo che sia completamente moltiplicativo anche nel "denominatore", e chiaramente c'è un unico modo di farlo, il seguente.

Se a è un intero ed n un intero positivo dispari, e scriviamo $n = p_1 \cdots p_r$, con p_1, \dots, p_r primi, non necessariamente distinti, definiamo il *simbolo di Jacobi*:

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right) \cdots \left(\frac{a}{p_r}\right).$$

Sarebbe possibile estendere la definizione ad un intero n qualsiasi, ma noi non ne avremo bisogno.

OSSERVAZIONI. Anche il simbolo di Jacobi, come quello di Legendre che esso estende, può valere ± 1 , ma se n è composto non è piú vero che a è un resto quadratico modulo n se e solo se $\left(\frac{a}{n}\right) = 1$. Quest'ultima è una condizione necessaria, ma non sufficiente, ad esempio 2 non è un resto quadratico modulo 15,

benché $\left(\frac{2}{15}\right) = \left(\frac{2}{3}\right)\left(\frac{2}{5}\right) = (-1)(-1) = 1$. Infatti se 2 fosse un resto quadratico modulo 15, lo dovrebbe essere anche modulo 3 e modulo 5, il che non è.

Più in generale, se $n = rs$, è un intero dispari con $(r, s) = 1$, allora un intero a primo con n è un resto quadratico modulo n se e solo se è un resto quadratico sia modulo r che modulo s (e nel caso in cui r, s siano primi questo avviene se e solo se $\left(\frac{a}{r}\right) = \left(\frac{a}{s}\right) = 1$). Infatti il verso (\Rightarrow) è ovvio; per (\Leftarrow) basta usare il teorema cinese dei resti, che da ciascuna coppia di soluzioni x_1 e x_2 per $x^2 \equiv a \pmod{r}$ e $x^2 \equiv a \pmod{s}$ permette di costruire le soluzioni del sistema

$$\begin{cases} x \equiv x_1 \pmod{r} \\ x \equiv x_2 \pmod{s} \end{cases}$$

e queste sono soluzioni di $x^2 \equiv a \pmod{rs}$.

Notate che (se n non è una potenza di un primo) i resti quadratici modulo n potranno anche essere meno di $\varphi(n)/2$, e quindi meno dei non-resti quadratici modulo n . D'altra parte se a è un resto quadratico modulo n , la congruenza $x^2 \equiv a \pmod{n}$ avrà in generale più di due soluzioni (pensate sempre modulo n).

Ad esempio se $n = pq$ con p, q primi dispari distinti, i resti quadratici modulo n saranno $\varphi(pq)/4$ ed i non-resti saranno $3\varphi(pq)/4$. D'altra parte se a (primo con n) è un resto quadratico modulo pq , allora la congruenza $x^2 \equiv a \pmod{pq}$ avrà esattamente quattro soluzioni modulo pq .⁹

PROPOSIZIONE 2.27. *Per n intero positivo dispari abbiamo*

$$\left(\frac{-1}{n}\right) = (-1)^{(n-1)/2}, \quad \left(\frac{2}{n}\right) = (-1)^{(n^2-1)/8}.$$

DIMOSTRAZIONE. Le due uguaglianze valgono per n primo dispari. Inoltre i loro primi membri sono funzioni moltiplicative di n per definizione del simbolo di Jacobi, e si verifica che lo sono anche i secondi membri. Ad esempio, nella dimostrazione del Teorema 2.13 abbiamo già verificato che il secondo membro della seconda formula è funzione moltiplicativa di n , cioè che la mappa

$$\begin{array}{ccc} 1 + 2\mathbb{Z} & \rightarrow & \{\pm 1\} \\ n & \mapsto & (-1)^{(n^2-1)/8} \end{array}$$

è un omomorfismo di monoidi. Per la prima formula la verifica è analoga. \square

PROPOSIZIONE 2.28. *Per n, m interi positivi dispari vale*

$$\left(\frac{m}{n}\right) = (-1)^{(m-1)(n-1)/4} \left(\frac{n}{m}\right).$$

DIMOSTRAZIONE. Possiamo assumere $(m, n) = 1$, perché in caso contrario entrambi i membri valgono 0. Scriviamo $m = p_1 \dots p_r$ e $n = q_1 \dots q_s$ come prodotti

⁹Un modo di vederlo è notare che

$$U(\mathbb{Z}/pq\mathbb{Z}) \cong U(\mathbb{Z}/p\mathbb{Z}) \times U(\mathbb{Z}/q\mathbb{Z}) \cong C_{p-1} \times C_{q-1}$$

il prodotto di due gruppi ciclici entrambi di ordine pari; il suo endomorfismo dato da $x \mapsto x^2$ ha nucleo di ordine 4 e quindi immagine di indice 4.

di primi, non necessariamente distinti. Abbiamo

$$\left(\frac{m}{n}\right) = \prod_{i,j} \left(\frac{p_i}{q_j}\right) = \pm \prod_{i,j} \left(\frac{q_j}{p_i}\right) = \pm \left(\frac{n}{m}\right),$$

dove il segno è dato da

$$\prod_j \left(\prod_i (-1)^{(p_i-1)/2}\right)^{(q_j-1)/2} = ((-1)^{(m-1)/2})^{(n-1)/2},$$

grazie alla moltiplicatività di $n \mapsto (-1)^{(n-1)/2}$, già usata nella dimostrazione precedente. \square

OSSERVAZIONI. La moltiplicatività di $n \mapsto (-1)^{(n-1)/2}$, per n dispari, usata nella precedente ed altre dimostrazioni, equivale chiaramente al fatto che la mappa

$$\begin{aligned} 1 + 2\mathbb{Z} &\rightarrow \mathbb{Z}/2\mathbb{Z} \\ n &\mapsto (n-1)/2 \end{aligned}$$

sia un omomorfismo del monoide moltiplicativo $1 + 2\mathbb{Z}$ nel gruppo additivo $\mathbb{Z}/2\mathbb{Z}$. In un certo senso quindi questa mappa si comporta come una specie di logaritmo. Questo semplice fatto ammette la seguente importante generalizzazione. Grazie al Teorema di Eulero-Fermat, se p è un primo ed a è un intero non multiplo di p , allora $a^{p-1} - 1$ è multiplo di p , e quindi $q_p(a) := (a^{p-1} - 1)/p$ è un intero, detto il *quoziente di Fermat* di a rispetto al primo p . Esso ha in passato giocato un ruolo importante in studi sull'*Ultimo teorema di Fermat*. È immediato verificare che per a e b interi non multipli di p vale $q_p(ab) \equiv q_p(a) + q_p(b)$. In altre parole, la funzione q_p è un omomorfismo del monoide moltiplicativo costituito dagli interi non multipli di p , nel gruppo additivo $\mathbb{Z}/p\mathbb{Z}$.

ESEMPIO 2.29. Possiamo calcolare un simbolo di Legendre come un simbolo di Jacobi e quindi non è più necessario scomporre in fattori, a parte separare potenze di 2 che è facile:

$$\begin{aligned} \left(\frac{1334}{1999}\right) &= \left(\frac{2}{1999}\right) \left(\frac{667}{1999}\right) = \left(\frac{667}{1999}\right) = - \left(\frac{1999}{667}\right) = - \left(\frac{-2}{667}\right) \\ &= - \left(\frac{-1}{667}\right) \left(\frac{2}{667}\right) = -(-1)(-1) = -1 \end{aligned}$$

senza bisogno di fattorizzare $667 = 23 \cdot 29$.

Notiamo che il costo computazionale del calcolo del simbolo di Jacobi con denominatore p è sostanzialmente quello per trovare il massimo comun divisore fra i suoi due argomenti, dunque $O(\log^2 p)$. Notate il miglioramento (anche se non drastico) rispetto all'utilizzo della Proposizione di Eulero, che portava ad una complessità $O(\log^3 p)$.

2.3. Estrazione di radici quadrate modulari

Segue facilmente dal Teorema cinese dei resti che un intero a è un resto quadratico modulo un intero $n = p_1^{\alpha_1} \cdots p_r^{\alpha_r}$ (con $p_1 < \cdots < p_r$ primi) se e solo se lo è modulo ciascuna potenza $p_i^{\alpha_i}$. Inoltre, se a è un tale resto quadratico, il

Teorema cinese dei resti permette di ridurre il calcolo di tutte le radici quadrate di a modulo n (che possono essere piú di due nel caso generale, come sappiamo) al calcolo di radici quadrate modulo p^α . In questa sezione vedremo degli algoritmi efficienti per risolvere quest'ultimo problema, iniziando con l'estrazione di radici quadrate modulo un primo.

2.3.1. Radici quadrate modulo p . Sia p un primo dispari ed a un intero con p che non divide a , che sia un resto quadratico modulo p , cioè tale che $\left(\frac{a}{p}\right) = 1$ (vale a dire $a^{(p-1)/2} \equiv 1 \pmod{p}$). Vediamo come si fa a calcolare le due radici quadrate di a modulo p , cioè le soluzioni di $x^2 \equiv a \pmod{p}$.

Anzitutto, c'è un modo semplice che funziona per metà dei primi, e precisamente se $p \equiv 3 \pmod{4}$: una radice quadrata di a modulo p è data da $r = a^{(p+1)/4} \pmod{p}$ (e l'altra è ovviamente il suo opposto). In effetti, $r^2 \equiv a^{(p+1)/2} \equiv a \cdot a^{(p-1)/2} \equiv a \pmod{p}$.

Anche per la metà dei primi rimanenti, precisamente per $p \equiv 5 \pmod{8}$, c'è una soluzione abbastanza semplice, ma non così immediata. Essendo $a^{(p-1)/2} \equiv 1 \pmod{p}$ avremo $a^{(p-1)/4} \equiv \pm 1 \pmod{p}$. Se il segno è positivo, è facile verificare in modo analogo al caso precedente che $r = a^{(p+3)/8} \pmod{p}$ soddisfa $r^2 \equiv a \pmod{p}$. Invece, se il segno è negativo, sarà $r^2 \equiv -a \pmod{p}$; poco male, infatti possiamo sfruttare il fatto che $2^{(p-1)/2} \equiv -1 \pmod{p}$, per cui $2^{(p-1)/4}$ è una radice quadrata di -1 modulo p , e quindi $2^{(p-1)/4} a^{(p+3)/8} = 2a \cdot (4a)^{(p-5)/8}$ è una radice quadrata di a modulo p .

Notate che in tutti i casi visti, se assumiamo che a sia già ridotto modulo p , diciamo $0 \leq a < p$, ed utilizziamo il metodo dei quadrati ripetuti, una radice di a modulo p si calcola in $O(\log^3 p)$ operazioni bit. Rimane da risolvere il caso $p \equiv 1 \pmod{8}$. Sarebbe nuovamente possibile risolvere mediante formule esplicite alcuni sottocasi (ad esempio, a seconda della classe di congruenza di p modulo 24), ma uno di questi casi andrebbe nuovamente diviso in sottocasi, e così via, quindi questa non è una strada praticabile.

Presentiamo ora un algoritmo, dovuto a Tonelli e Shanks, per calcolare una radice quadrata di a modulo p , che è la generalizzazione naturale dei casi particolari appena esaminati. Dobbiamo però assumere di avere a disposizione un non-resto quadratico n modulo p . Nell'esporre l'algoritmo, metteremo fra parentesi quadre le ragioni di teoria dei gruppi che giustificano le operazioni eseguite. Iniziamo scrivendo $p-1 = 2^\alpha \cdot s$ con s dispari.

[Dunque $U(\mathbb{Z}/p\mathbb{Z}) \cong C_{2^\alpha} \times C_s$ (ad esempio grazie al Teorema cinese dei resti). In ogni gruppo G di ordine dispari s la mappa $g \mapsto g^2$ è biettiva e la sua inversa si può scrivere esplicitamente come $g \mapsto g^{(s+1)/2}$, infatti $(g^{(s+1)/2})^2 = g^s \cdot g^1 = g$. Quindi l'unica radice quadrata di g in G è $g^{(s+1)/2}$. (Ad esempio, nel caso speciale $p \equiv 3 \pmod{4}$ visto all'inizio abbiamo $p-1 = 2s$ con s dispari, ed essendo a un resto quadratico, a appartiene al sottogruppo di indice due di \mathbb{F}_p^* , per cui $a^{(s+1)/2} = a^{(p+1)/4}$ è già una radice quadrata di a modulo p .) Nella nostra situazione, facendoci guidare dall'isomorfismo visto porremo

$$r := (a_1, a_2)^{(s+1)/2} = \left(a_1^{(s+1)/2}, a_2^{(s+1)/2} \right)$$

Dunque $a_2^{(s+1)/2}$ sarà l'unica radice quadrata di a_2 in C_s ; perciò il rapporto fra r e una radice quadrata di a modulo p sarà un elemento di C_{2^α} , quindi di ordine una potenza di 2. In altre parole, il quoziente $a^{-1}r^2 = a_1^s$ sarà un quadrato in C_{2^α} , poiché a_2 lo era, essendo a un resto quadratico modulo p . Quindi l'ordine di $a^{-1}r^2$ dividerà $2^{\alpha-1}$. Ora lo verifichiamo direttamente, senza ricorrere all'isomorfismo col prodotto diretto.]

Iniziamo dunque calcolando $r := a^{(s+1)/2}$. Essendo

$$(a^{-1}r^2)^{2^{\alpha-1}} = a^{s2^{\alpha-1}} = a^{(p-1)/2} = \left(\frac{a}{p}\right) = 1$$

(dove a^{-1} indica un inverso di a modulo p) vediamo che l'ordine di $a^{-1}r^2$ modulo p divide $2^{\alpha-1}$. Dunque possiamo considerare r una prima approssimazione di una radice quadrata di a modulo p . (Se per caso tale ordine è 1, allora r è *davvero* una radice quadrata di a modulo p .) L'idea è quella di ottenere approssimazioni via via migliori modificando r : il singolo passo, che vedremo fra un attimo, da ripetere eventualmente varie volte, è moltiplicare r per un'opportuna radice 2^α -esima dell'unità in \mathbb{F}_p . Ma prima ci serve una 2^α -esima primitiva dell'unità in \mathbb{F}_p .

Abbiamo supposto di avere a disposizione un non-resto quadratico n modulo p . Ne otteniamo una radice 2^α -esima primitiva dell'unità $b = n^s$; infatti, 2^α divide $|n|$, e quindi $|n^s| = |n|/(|n|, s) = 2^\alpha$. Che b abbia ordine 2^α segue anche dal calcolo

$$b^{2^{\alpha-1}} = n^{s2^{\alpha-1}} = n^{(p-1)/2} = \left(\frac{n}{p}\right) = -1.$$

[Oppure, si può notare che nell'isomorfismo introdotto in precedenza $b = n^s$ corrisponde a $(n_1, n_2)^s = (n_1^s, n_2^s) = (n_1^s, 1)$, ed essendo s dispari n_1^s ha lo stesso ordine di n_1 , che è 2^α in quanto n_1 non è un quadrato in C_{2^α} . A questo punto sappiamo che $a^{-1}r^2 \in \mathbb{F}_p^*$ appartiene all'unico sottogruppo di \mathbb{F}_p^* di ordine $2^{\alpha-1}$, che è generato da b^2 . Quindi r differisce da una vera radice quadrata di a modulo p per un fattore moltiplicativo che è un'opportuna potenza b^j . Si possono calcolare le cifre dell'espansione binaria dell'esponente j come suggerisce [Kob94], o procedere nel seguente modo equivalente.]

Supponiamo che $a^{-1}r^2$ abbia ordine 2^β con $0 < \beta < \alpha$, il che equivale ad $(a^{-1}r^2)^{2^{\beta-1}} = -1$. (Dunque in pratica possiamo calcolare β elevando $a^{-1}r^2$ ripetutamente al quadrato fino ad ottenere -1 .) Avremo

$$\left(a^{-1} \left(rb^{2^{\alpha-\beta-1}}\right)^2\right)^{2^{\beta-1}} = -b^{2^{\alpha-1}} = 1$$

e quindi rimpiazzando r con $r' = rb^{2^{\alpha-\beta-1}}$ otterremo un'approssimazione migliore r' di una radice quadrata di a (mod p). Ripetendo quest'ultimo passo (stavolta con r' al posto di r) un numero sufficiente di volte (al più $\beta - 1$) otterremo una radice quadrata di a (mod p).

[Quest'ultimo passaggio si può giustificare nel modo seguente: se g ed h sono elementi di un gruppo ciclico ed hanno ordine la stessa potenza di 2, diciamo 2^β ,

allora il loro prodotto gh ha ordine minore di 2^β ; infatti g ed h generano lo stesso sottogruppo, di ordine 2^β , perciò $h = g^t$ con t dispari, e quindi $gh = g^{t+1}$, con $t + 1$ pari.]

OSSERVAZIONI. Il tempo richiesto dall'algoritmo è $O(\log^4 p)$. Infatti, a parte il calcolo iniziale di $r = a^{(s+1)/2}$, il passo da ripetere al più $\alpha - 1$ volte, cioè $O(\log p)$ volte, è calcolare l'ordine di $a^{-1}r^2$ (e poi $a^{-1}(r')^2$, ecc.), che richiede $O(\log^3 p)$ operazioni bit (più altre operazioni meno costose, come il calcolo delle opportune potenze b^{2^i} , che complessivamente costa $O(\log^3 p)$ operazioni bit).

OSSERVAZIONI. L'algoritmo come descritto è deterministico (così come le formule per i casi particolari visti all'inizio), non appena conosciamo un non-resto quadratico n modulo p . (Si veda a tal proposito anche l'Osservazione successiva.) Scoprire un tale n non è in pratica difficile per tentativi calcolando vari simboli di Legendre: farlo in questo modo però è un algoritmo efficiente, ma probabilistico, nel senso seguente. La probabilità di non avere ancora scoperto un non-resto quadratico modulo p dopo i tentativi è al massimo 2^{-i} ; quindi se fissiamo $\varepsilon > 0$ possiamo limitare il numero di tentativi da eseguire a $\lfloor -\log_2 \varepsilon \rfloor$, e l'algoritmo risultante finirà in $O(\log^2 p)$ operazioni bit (il tempo per calcolare un numero costante di simboli di Legendre), ma non avrà successo garantito, bensì con probabilità maggiore di $1 - \varepsilon$. Naturalmente esistono algoritmi deterministici per trovare un non-resto quadratico, ad esempio quello che calcola il simbolo di Legendre $\left(\frac{c}{p}\right)$ per ogni c compreso fra 2 e $p - 1$ (anzi, basterebbero la metà più uno) ma il tempo impiegato non è polinomiale, ma soltanto $O(p \log^2 p)$.

OSSERVAZIONI. A proposito della difficoltà, nominata poco fa, di trovare un non-resto quadratico modulo p in modo deterministico efficiente, notiamo che la soluzione più semplice a cui uno potrebbe pensare non funziona: non esiste un intero n che sia un non-resto quadratico modulo tutti i primi dispari. Anzi, non esiste nemmeno un intero n che sia un non-resto quadratico modulo p per ogni $p \equiv 1 \pmod{8}$. (Notate che invece: -1 e -2 sono non-resti quadratici modulo ogni $p \equiv -1 \pmod{8}$; -1 e 2 sono non-resti quadratici modulo ogni $p \equiv 3 \pmod{8}$; 2 e -2 sono non-resti quadratici modulo ogni $p \equiv -3 \pmod{8}$.)

Per mostrarlo, fissiamo un intero n . Mostreremo che esiste almeno un primo, anzi, esistono infiniti primi $p \equiv 1 \pmod{8}$ tali che n è un resto quadratico modulo p . Anzitutto scriviamo $n = \varepsilon 2^k d$ con $\varepsilon = \pm 1$ e d un intero positivo dispari. Poiché sia -1 che 2 sono quadrati modulo p , qualunque sia $p \equiv 1 \pmod{8}$ con $p \nmid n$ avremo $\left(\frac{n}{p}\right) = \left(\frac{d}{p}\right) = \left(\frac{p}{d}\right)$. A questo punto sarà sufficiente trovare un primo $p \equiv 1 \pmod{8}$ tale che p sia un resto quadratico modulo d , e questa seconda condizione è soddisfatta ad esempio se $p \equiv 1 \pmod{d}$. Concludiamo mettendo a sistema le due congruenze ed applicando all'insieme delle soluzioni il Teorema di Dirichlet sui primi in progressione aritmetica.

Prima di concludere con un esempio, notiamo che l'algoritmo di Tonelli e Shanks è essenzialmente un caso speciale del logaritmo di Silver-Pohlig-Hellman esposto nella Sottosezione 3.3.4. La ragione è che, almeno formalmente, estrarre

una radice quadrata equivale ad estrarre un logaritmo, dimezzarlo e quindi calcolare l'esponentiale del risultato. Infatti, riprendendo la notazione usata in precedenza, l'ordine modulo p di a^s è un divisore di 2^α , anzi di $2^{\alpha-1}$ avendo assunto che a sia un resto quadratico, e quindi appartiene al sottogruppo di ordine 2^α di $U(\mathbb{Z}/p\mathbb{Z})$, che è generato da $n^s = b$, anzi, al sottogruppo generato da b^2 . Ma allora $a^{-s} = b^x$ per un intero x (unico modulo 2^α), e x è pari, perciò $b^{x/2}$ è una radice quadrata di a^{-s} modulo p , e quindi $a^{(s+1)/2}b^{x/2}$ è una radice quadrata di a modulo p . Scrivendo $x = 2x_1 + 4x_2 + \dots + 2^{\alpha-1}x_{\alpha-1} \pmod{2^\alpha}$ e procedendo come nell'algoritmo di Silver-Pohlig-Hellman possiamo determinare x_1, x_2, \dots , ed infine calcolare $b^{x/2}$. Il numero β trovato al primo passo dell'algoritmo di Tonelli e Shanks (cioè tale che $a^{-1}r^2 = a^s$ abbia ordine 2^β modulo p) corrisponde qui al fatto che $x_{\alpha-\beta}$ sia il bit meno significativo non nullo nella scrittura binaria di x .

ESEMPIO 2.30. Calcoliamo la radice quadrata di $a = 47$ modulo il primo $241 = 2^4 \cdot 15 + 1$. Una "prima approssimazione" della radice quadrata di 47 modulo 241 è perciò $r := 47^{(15+1)/2} \equiv 98 \pmod{241}$. Abbiamo che $a^{-1} \equiv -41 \pmod{241}$, $r^2 \equiv -36 \pmod{241}$, e quindi $a^s = a^{-1}r^2 \equiv 30 \not\equiv 1 \pmod{241}$. Dunque r non è una radice quadrata di 47.

A questo punto per proseguire ci serve un non-resto quadratico modulo 241; si verifica che 2, 3 e 5 sono resti quadratici, mentre 7 non lo è, quindi scegliamo $n = 7$. Ne otteniamo l'elemento $b := n^{15} \equiv 111 \pmod{241}$ di ordine 16 modulo 241. Non è indispensabile, ma controlliamo per sicurezza:

$$b^2 \equiv 111^2 \equiv 30, \quad b^4 \equiv (-64)^2 \equiv -64, \quad b^8 \equiv 30^2 \equiv -1.$$

Quindi $a^s = a^{-1}r^2 \equiv 30$ deve essere una potenza di b con esponente pari. (In effetti, il conto appena fatto mostra che $a^s \equiv b^2$, e quindi $rb^{-1} = a^{(s+1)/2}b^{-1} \equiv -23$ è una radice quadrata di $a = 47$; ignoriamolo, per illustrare l'algoritmo.) Essendo ($30^2 \equiv -64$, e poi) $30^4 \equiv -1$ abbiamo che 30 ha ordine 8 (il massimo possibile a questo punto). Perciò rimpiazziamo r con $r' = rb \equiv 33$, che deve essere una migliore approssimazione della radice quadrata di a , rispetto a quella iniziale r . Abbiamo $a^{-1}(r')^2 = a^{-1}r^2b^2 \equiv -64 \not\equiv 1$, quindi non abbiamo ancora trovato una radice quadrata di a , e dobbiamo proseguire.

Ora $a^{-1}(r')^2 \equiv -64$ deve essere una potenza di b , con esponente multiplo di 4. (Di nuovo, se ci accorgessimo di aver già incontrato -64 come b^4 scopriremmo che $a = (r')^2b^{-4} = r^2b^{-2}$, da cui $rb^{-1} \equiv -23$ è una radice quadrata di 47, come già notato; ignoriamo anche questo per fare pratica.) Dobbiamo dunque calcolare l'ordine di $a^{-1}(r')^2 \equiv -64$, che sappiamo dividere 4. Essendo $(-64)^2 \equiv -1$, tale ordine è proprio 4. Perciò rimpiazziamo r' con $r'' = r'b^2 = rb^3 \equiv 26$. Ora abbiamo $a^{-1}(r'')^2 = a^{-1}r^2b^6 \equiv -1$, che ha ordine 2, e quindi il passo finale è porre $r''' = r''b^4 = rb^7 \equiv 23$. Questa è una radice quadrata di 47 modulo 241.

2.3.2. Resti quadratici modulo p^α . Vediamo ora di capire quali sono i resti quadratici modulo una potenza p^α di un primo p . Se $p > 2$ la situazione è molto semplice: un intero a non multiplo di p (altrimenti è facile) è un resto quadratico modulo p^α se e solo se è un resto quadratico modulo p . Un verso è immediato, mentre l'altro segue dall'algoritmo per *sollevare* una radice quadrata modulo p ad una modulo p^α che descriviamo nella prossima sezione.

È comunque istruttivo darle una dimostrazione piú teorica (e forse piú elegante) usando un po' di teoria dei gruppi. Intanto, $U(\mathbb{Z}/p^\alpha\mathbb{Z})$ è ciclico, come abbiamo visto nel capitolo precedente, e dunque 1 e -1 sono gli unici elementi di tale gruppo che al quadrato fanno 1. Come nel caso $\alpha = 1$, segue che ci sono $\varphi(p^\alpha)/2$ resti quadratici modulo p^α , e altrettanti non-resti quadratici. Ora la riduzione modulo p è un omomorfismo $\psi : U(\mathbb{Z}/p^\alpha\mathbb{Z}) \rightarrow U(\mathbb{Z}/p\mathbb{Z})$ che manda ovviamente quadrati in quadrati. D'altra parte le immagini inverse in $U(\mathbb{Z}/p^\alpha\mathbb{Z})$ dei quadrati di $U(\mathbb{Z}/p\mathbb{Z})$ sono in numero di $p^{\alpha-1} \cdot (p-1)/2 = \varphi(p^\alpha)/2$, cioè tante quanti i quadrati in $U(\mathbb{Z}/p^\alpha\mathbb{Z})$. Ne segue che tutte le immagini inverse di quadrati in $U(\mathbb{Z}/p\mathbb{Z})$ sono quadrati in $U(\mathbb{Z}/p^\alpha\mathbb{Z})$.¹⁰

Applicando il Teorema cinese dei resti, possiamo anche concludere che un intero a primo con n dispari è un resto quadratico modulo n sse è un resto quadratico modulo ogni divisore primo di n .

Se $p = 2$ la situazione è piú complicata: per sapere se un intero (dispari, altrimenti è facile) a è un resto quadratico modulo 2^α (con $\alpha > 3$) non è sufficiente sapere se è un resto quadratico modulo 2 (che è banalmente sempre vero), ma bisogna sapere se è un resto quadratico modulo 8.¹¹ Infatti, gli interi dispari che sono resti quadratici modulo 2^α , con $\alpha > 3$, sono esattamente gli interi congrui a 1 modulo 8.

Come nel caso dispari un verso è banale, mentre il verso opposto segue dall'algoritmo descritto nella prossima sezione, ma anche qui ne vogliamo dare una dimostrazione indipendente. Sappiamo che $U(\mathbb{Z}/2^\alpha\mathbb{Z})$ non è ciclico, ma è il prodotto diretto di un gruppo ciclico di ordine 2 per uno di ordine $2^{\alpha-2}$. Ne segue che esso ha esattamente quattro elementi di ordine che divide 2, come abbiamo visto nel capitolo precedente. Di conseguenza, i resti quadratici modulo 2^α sono in numero di $\varphi(2^\alpha)/4 = 2^{\alpha-2}$. (D'altra parte, gli elementi di $U(\mathbb{Z}/2^\alpha\mathbb{Z})$ che sono quadrati, hanno *quattro* radici quadrate distinte.) Stavolta consideriamo l'omomorfismo $U(\mathbb{Z}/2^\alpha\mathbb{Z}) \rightarrow U(\mathbb{Z}/8\mathbb{Z})$ dato dalla riduzione modulo 8. Elementi di $U(\mathbb{Z}/2^\alpha\mathbb{Z})$ che sono quadrati vengono ovviamente mandati in quadrati di $U(\mathbb{Z}/8\mathbb{Z})$, e dunque appartengono al nucleo dell'omomorfismo. Ma il nucleo ha $\varphi(2^\alpha)/4$ elementi, tanti quanti i quadrati in $U(\mathbb{Z}/2^\alpha\mathbb{Z})$, e quindi tutti gli elementi del nucleo sono quadrati.¹²

¹⁰Volendo possiamo evitare di usare il fatto, comunque fondamentale, che il gruppo $U(\mathbb{Z}/p^\alpha\mathbb{Z})$ è ciclico, nel modo seguente: se $x^2 \equiv 1 \pmod{p^\alpha}$, allora $p^\alpha \mid (x-1)(x+1)$, da cui $p \mid x-1$ o $p \mid x+1$, essendo p primo; essendo $p > 2$, ciascuna delle due conclusioni esclude l'altra, e quindi avremo che $p^\alpha \mid x-1$ o $p^\alpha \mid x+1$, e dunque $x \equiv \pm 1 \pmod{p^\alpha}$.

¹¹Ricordo che 1 è l'unico resto quadratico dispari modulo 8, per verifica diretta (cioè $(\pm 1)^2 \equiv 1$ e $(\pm 3)^2 \equiv 1$), o anche nel modo seguente, forse piú illuminante: $(2k+1)^2 = 4k(k+1) + 1$, e uno fra k e $k+1$ deve essere pari.

¹²In effetti, del teorema dove abbiamo descritto la struttura di $U(\mathbb{Z}/p^\alpha\mathbb{Z})$ ci serve solo il seguente passo della dimostrazione: se $x^2 \equiv 1 \pmod{2^\alpha}$, allora $2^\alpha \mid (x-1)(x+1)$, e quindi $x-1$ e $x+1$ sono entrambi pari, ma solo uno dei due sarà multiplo di quattro; ne segue che quello dovrà anzi essere multiplo di $2^{\alpha-1}$, e quindi $x \equiv \pm 1, 2^{\alpha-1} \pm 1 \pmod{2^\alpha}$; d'altra parte, il quadrato di ognuno di tali interi è multiplo di 2^α .

2.3.3. Radici quadrate modulo p^α . Se p è un primo dispari e $\left(\frac{a}{p}\right) = 1$, allora $x^2 \equiv a \pmod{p}$ ha soluzione e per quanto visto ha soluzione anche ogni $x^2 \equiv a \pmod{p^\alpha}$, per ogni α . Per calcolare una tale soluzione volendo si può utilizzare lo stesso metodo di Tonelli e Shanks utilizzato in precedenza per estrarre le radici quadrate modulo p : infatti il metodo non sfruttava tanto il fatto che p fosse primo, quanto il fatto che $U(\mathbb{Z}/p\mathbb{Z})$ fosse ciclico. Essendo anche $U(\mathbb{Z}/p^\alpha\mathbb{Z})$ ciclico, stavolta di ordine $p^{\alpha-1}(p-1)$, il metodo inizierà scrivendo $p^{\alpha-1}(p-1) = 2^{\beta \cdot s}$ con s dispari, ecc.; ad un certo punto servirà conoscere un non-resto quadratico n modulo p^α , che è la stessa cosa di un non-resto quadratico modulo p .

Per quanto osservato in precedenza, la complessità del metodo (avendo già a disposizione n) sarà di $O(\log^4(p^\alpha)) = O(\alpha^4 \log^4 p)$ operazioni bit. Possiamo però fare molto meglio. Infatti, se già conosciamo una radice quadrata di a modulo p , il metodo che stiamo per vedere ci permette di *sollevarla* ad una radice quadrata modulo p^α (vale a dire, una radice modulo p^α che sia congrua modulo p alla radice data) in solo $O(\alpha^2 \log^2 p)$ operazioni bit (come potete verificare per esercizio). Quindi, anche se non conosciamo una radice quadrata di a modulo p , conviene prima calcolarla (in $O(\log^4 p)$ operazioni bit), e poi sollevarla ad una modulo p^α (specialmente se α è grande rispetto a $\log p$). Alla fine vedremo che una leggera modifica al metodo permette anche di estrarre le radici quadrate modulo 2^α .

Il metodo funziona per “approssimazioni successive”. Supponiamo di avere x_α tale che $x_\alpha^2 \equiv a \pmod{p^\alpha}$, per qualche $\alpha \geq 1$. (Dunque le soluzioni di tale congruenza sono tutti e solo gli interi congrui a $\pm x_\alpha$ modulo p^α .) Una eventuale soluzione $x_{\alpha+1}$ di $x^2 \equiv a \pmod{p^{\alpha+1}}$ (che in realtà sappiamo già esistere, ma se lo ignoriamo stiamo per dare un'altra dimostrazione della sua esistenza) dovrà necessariamente essere congrua a $\pm x_\alpha$ modulo p^α . Diciamo ad esempio $x_{\alpha+1} = x_\alpha + tp^\alpha$ per qualche t . Allora deve essere $(x_\alpha + tp^\alpha)^2 \equiv a \pmod{p^{\alpha+1}}$. Quindi $x_\alpha^2 + 2tx_\alpha p^\alpha \equiv a \pmod{p^{\alpha+1}}$, e

$$tp^\alpha \equiv \frac{a - x_\alpha^2}{2x_\alpha} \pmod{p^{\alpha+1}}$$

Attenzione: abbiamo scritto una divisione per $2x_\alpha$, ma in realtà intendiamo moltiplicare il numeratore per un inverso di $2x_\alpha$ modulo p^α , che esiste in quanto p è dispari e non divide a . Notate anche che il numeratore è per ipotesi multiplo di p^α , dunque la congruenza ammette soluzioni per t , individuate da

$$t \equiv \frac{a - x_\alpha^2}{p^\alpha \cdot 2x_\alpha} \pmod{p}.$$

Con una tale scelta di t ottengo una soluzione $x_{\alpha+1}$ di $x_{\alpha+1}^2 \equiv a \pmod{p^{\alpha+1}}$ data da

$$x_{\alpha+1} = x_\alpha + \frac{a - x_\alpha^2}{2x_\alpha} = \frac{x_\alpha + \frac{a}{x_\alpha}}{2} \pmod{p^{\alpha+1}}$$

(che in pratica possiamo usare per costruire direttamente $x_{\alpha+1}$ a partire da x_α , senza passare per il calcolo di t).

Possiamo anche vedere l'algoritmo in un altro modo. Ripartiamo da capo. Ricordo il metodo delle tangenti di Newton. Sia $f : \mathbb{R} \rightarrow \mathbb{R}$ continua e derivabile con derivata continua. Se cerchiamo soluzioni di $f(x) = 0$, possiamo partire da un valore x_1 e poi trovare approssimazioni successive mediante la formula

$$x_{\alpha+1} = x_\alpha - \frac{f(x_\alpha)}{f'(x_\alpha)}.$$

[Se non conoscete il metodo di Newton, fate un disegno di come ricavate $x_{\alpha+1}$ da x_α e capirete perché si chiama *metodo delle tangenti*.] Sotto opportune condizioni questa successione converge ed in tal caso, se ξ è il suo limite, riscrivendo la formula nella forma

$$f(x_\alpha) = f'(x_\alpha) \cdot (x_\alpha - x_{\alpha+1})$$

e passando al limite otteniamo che $f(\xi) = 0$.

Il metodo di Newton funziona anche per sollevare una radice (intera) di un polinomio $f(x) \equiv 0 \pmod{p^\alpha}$ ad una di $f(x) \equiv 0 \pmod{p^{\alpha+1}}$, purché $f'(x_\alpha) \not\equiv 0 \pmod{p}$. In particolare, se x_α è una soluzione di $x^2 \equiv a \pmod{p^\alpha}$, con $(a, p) = 1$, allora

$$x_{\alpha+1} := x_\alpha - \frac{x_\alpha^2 - a}{2x_\alpha} = \frac{x_\alpha + \frac{a}{x_\alpha}}{2}$$

è soluzione di $x^2 \equiv a \pmod{p^{2\alpha}}$ (e non solo modulo $p^{\alpha+1}$). Qui per dividere per x_α devo moltiplicare per un inverso di x_α sufficientemente preciso, cioè modulo $p^{2\alpha}$ (cioè nell'anello dove cerco il risultato). Infatti

$$\begin{aligned} x_{\alpha+1}^2 &= \left(\frac{x_\alpha^2 + a}{2x_\alpha} \right)^2 - a = \frac{x_\alpha^4 + 2ax_\alpha^2 + a^2}{4x_\alpha^2} - a = \frac{x_\alpha^4 - 2ax_\alpha^2 + a^2}{4x_\alpha^2} = \frac{(x_\alpha^2 - a)^2}{4x_\alpha^2} \\ &\equiv 0 \pmod{p^{2\alpha}} \end{aligned}$$

in quanto $p^\alpha \mid x_\alpha^2 - a$, e $p \nmid 4x_\alpha^2$. Osserviamo anche che $x_{\alpha+1} \equiv x_\alpha \pmod{p^\alpha}$, cioè $x_{\alpha+1}$ solleva proprio x_α , e non un'altra radice.

Da quest'ultima discussione si vede che la "convergenza" del metodo è più rapida di quanto apparisse dalla prima analisi, nel senso che con un singolo passo si va da una radice modulo p^α ad una modulo $p^{2\alpha}$, anziché soltanto modulo $p^{\alpha+1}$. Questo porta la complessità del metodo da $O(\alpha^3 \log^2 p)$, come poteva apparire dalla prima analisi, a $O(\alpha^2 \log^2 p)$ (verificatelo per esercizio).¹³

Vediamo ora brevemente le modifiche da apportare per trattare il caso $p = 2$. Supponiamo di avere x_α dove $\alpha \geq 3$, tale che $x_\alpha^2 \equiv a \pmod{2^\alpha}$, con $a \equiv 1$

¹³La similitudine fra l'applicazione del metodo di Newton ai reali (o volendo ai complessi, per f olomorfa) ed agli interi modulo p^α diverrebbe ancora più forte se introducessimo il campo \mathbb{Q}_p dei numeri p -adici, che sono il completamento (come spazio metrico) dei numeri razionali rispetto ad una certa distanza, la distanza p -adica, così come i reali sono il completamento dei razionali rispetto alla distanza ordinaria. La successione infinita degli x_α costruiti come visto, come radici quadrate di a modulo p^α con α crescente indefinitamente sarebbe una successione di approssimazioni sempre migliori di un certo numero p -adico, una delle radici quadrate di a in \mathbb{Q}_p (ovvero la successione convergerebbe a quel numero p -adico, in senso p -adico, mentre potrebbe anche divergere nel senso ordinario).

(mod 8) (che sappiamo essere condizione necessaria per la risolubilità della congruenza, sempre restringendoci al caso $2 \nmid a$). Mostriamo che esiste un intero t (che ovviamente possiamo sempre scegliere fra $t = 0, 1$) tale che $x_{\alpha+1} = x_{\alpha} + t2^{\alpha-1} \equiv x_{\alpha} \pmod{2^{\alpha-1}}$ (attenzione all'esponente $\alpha - 1$, anziché α come nel caso p dispari) e $x_{\alpha+1}^2 \equiv a \pmod{2^{\alpha+1}}$, e quindi basterà (e bisognerà) scegliere

$$t \equiv \frac{a - x_{\alpha}^2}{2^{\alpha} \cdot x_{\alpha}} \equiv \frac{a - x_{\alpha}^2}{2^{\alpha}} \pmod{2},$$

essendo $x_{\alpha} \equiv 1 \pmod{2}$. Si potrà ad esempio scegliere $t = 0$ o 1 a seconda che $(a - x_{\alpha}^2)/2^{\alpha}$ sia pari o dispari.

Volendo, anche in questo caso potremmo anche porre semplicemente

$$x_{\alpha+1} = x_{\alpha} + \frac{a - x_{\alpha}^2}{2x_{\alpha}} = \frac{x_{\alpha} + \frac{a}{x_{\alpha}}}{2} \pmod{2^{\alpha+1}}.$$

14

¹⁴Il fatto che, a differenza del caso dispari, da una radice quadrata x_{α} di a modulo 2^{α} ne ricaviamo una modulo $2^{\alpha+1}$, e precisamente $x_{\alpha+1}$, congrua alla precedente soltanto modulo $2^{\alpha-1}$, è perché ad ogni singolo passaggio a ha *quattro* radici quadrate, ma solo *due* di esse si sollevano al passaggio successivo (cioè hanno a loro volta radici quadrate nel gruppo più grande).

CAPITOLO 3

Crittografia

3.1. La crittografia in generale

Un *sistema di crittografia* è rappresentabile in questo modo

$$\mathcal{P} \xrightarrow{f} \mathcal{C} \xrightarrow{f^{-1}} \mathcal{P}$$

dove

- \mathcal{P} = insieme dei possibili messaggi elementari,
- \mathcal{C} = insieme dei messaggi crittati,
- f = funzione che critta i messaggi,
- f^{-1} = funzione di cui abbiamo bisogno per riottenere il messaggio originale (notiamo che è sufficiente l'iniettività di f , non è necessario che sia invertibile, infatti basta prendere come f^{-1} una qualsiasi inversa destra di f ; per facilitare la descrizione del codominio \mathcal{C} sarà talvolta conveniente non dover richiedere che f sia suriettiva).

Ad esempio \mathcal{P} potrebbe essere costituito dalle lettere dell'alfabeto tradotte in forma numerica, che possono essere una lettera alla volta (incluso magari la punteggiatura, ad esempio i 256 caratteri ASCII) o blocchi di lettere in una volta (coppie di lettere, terne di lettere o blocchi di k lettere), oppure tradotte in elementi di un gruppo (che potrebbe essere il gruppo degli elementi non nulli di un campo finito, oppure il gruppo dei punti di una *curva ellittica* su un campo finito).

In un sistema di crittografia non è necessario che sia segreto il modo in cui le lettere sono associate ai numeri, ma deve essere segreta la funzione f (il segreto deve essere condiviso solo da colui che manda e colui che riceve il messaggio).

Si può pensare ad un sistema di crittografia come una specie di dizionario, che per ogni “parola” dell'insieme \mathcal{P} , che chiameremo più propriamente *unità di messaggio*, fornisce la “traduzione”, cioè l'equivalente crittato, che è una “parola” dell'insieme \mathcal{C} .

Un modo per aumentare la sicurezza di un sistema può essere quello di prendere \mathcal{P} grande (anziché prendere per \mathcal{P} le 26 lettere dell'alfabeto si possono prendere le 26^{10} sequenze di 10 lettere), e di scegliere f il più *casuale* possibile. Chiaramente queste esigenze contrastano con la necessità di mantenere il sistema effettivamente maneggiabile, quindi vanno fatti dei compromessi con la semplicità d'uso (operazioni fattibili per eseguire f), la rapidità di implementazione (o meglio di esecuzione), problemi di memoria, ecc. Per questa serie di motivi di solito si sceglie f all'interno di una classe di funzioni descrivibili in modo semplice.

La crittografia trova applicazioni nelle comunicazioni, nei telefonini (in modo che un terzo che intercetta le onde radio non possa capire cosa ci si sta dicendo), nelle carte di credito, ecc.

ESEMPIO 3.1. Vediamo un caso particolare (molto usato nell'antichità, ad esempio dai Romani). Siano \mathcal{P} e \mathcal{C} formati dalle 26 lettere dell'alfabeto inglese e sia f una funzione che trasla le lettere. La chiave consiste nel sapere di quante posizioni vengono traslate le lettere

$$\begin{array}{rcccccc} \mathcal{P} : & A & B & C & D & \dots & Z \\ \mathcal{C} : & E & F & G & H & \dots & D \end{array}$$

Possiamo generalizzare questo esempio prendendo $\mathcal{P} = \mathcal{C} = \mathbb{Z}/N\mathbb{Z}$ e $f(P) = P + b \pmod{N}$, dove $P \in \mathcal{P}$ e b è un qualsiasi elemento di $\mathbb{Z}/N\mathbb{Z}$. In questo caso f è biiettiva e $f^{-1}(C) = C - b \pmod{N}$, $\forall C \in \mathcal{C}$. Prendiamo per esempio $N = 26$ e $b = 3$ e facciamo corrispondere ad ogni lettera il suo numero. Vogliamo crittare la parola CIAO, quindi a C corrisponde il numero 3, a I il numero 9, a A il numero 1 e a O il numero 15. Cioè prima di tutto associamo alla parola CIAO la stringa numerica 3 9 1 15 da crittare. Ora

$$\begin{aligned} f(3) &= 3 + 3 = 6 \\ f(9) &= 9 + 3 = 12 \\ f(1) &= 1 + 3 = 4 \\ f(15) &= 15 + 3 = 18 \end{aligned}$$

Quindi noi spediamo 6 12 4 18. Chi riceve il messaggio, usando f^{-1} riottiene la stringa iniziale

$$\begin{aligned} f^{-1}(6) &= 6 - 3 = 3 \\ f^{-1}(12) &= 12 - 3 = 9 \\ f^{-1}(4) &= 4 - 3 = 1 \\ f^{-1}(18) &= 18 - 3 = 15 \end{aligned}$$

ESEMPIO 3.2. È una generalizzazione del metodo precedente. Sia $\mathcal{P} = \mathcal{C} = \mathbb{Z}/N\mathbb{Z}$ e sia f la mappa $f(P) = aP + b \pmod{N}$, $\forall P \in \mathcal{P}$, dove $b \in \mathbb{Z}/N\mathbb{Z}$ e a è invertibile modulo N , cioè $a \in U(\mathbb{Z}/N\mathbb{Z})$. (Una tale f è detta *mappa affine* (invertibile) su $\mathbb{Z}/N\mathbb{Z}$) L'inversa di f è $f^{-1}(C) = a^{-1}C - a^{-1}b \pmod{N}$.

Notiamo che questo sistema è altamente vulnerabile in quanto si può attaccare con l'*analisi di frequenza*. L'operazione di un intruso che decodifica un messaggio segreto è detta *breaking* (o *violare* il codice in italiano) e la scienza che studia il modo di *decifrare i messaggi* non essendone autorizzati è detta *crittanalisi*. Di solito nella crittanalisi si assume che l'intruso conosca la forma generale del sistema di crittografia adottato, e che quindi l'unica cosa da scoprire sia la particolare chiave usata.

L'idea dell'analisi di frequenza è la seguente: supponiamo che un intruso abbia intercettato un messaggio sufficientemente lungo; egli può ragionevolmente supporre che certe lettere compaiono con una certa frequenza più di altre. Ad

esempio, in italiano la vocale e è una delle lettere che compaiono più di frequente, certo più della u o della q ; in generale, in ogni linguaggio si possono calcolare le percentuali con cui mediamente appaiono le varie lettere. La lettera che l'intruso vede comparire con maggiore frequenza nel messaggio crittato a sua disposizione corrisponderà probabilmente alla lettera in chiaro che compare con maggiore frequenza in quel linguaggio; analogamente con la seconda lettera, ecc. Naturalmente decrittando una lettera dopo l'altra in questo modo a un certo punto la probabilità di sbagliarsi diventerà inaccettabile. Tuttavia, la debolezza del sistema che stiamo esaminando è che non appena l'intruso conosce l'equivalente in chiaro di sole due lettere potrà ricavare b , f e f^{-1} , e diverrà quindi in grado di decrittare qualsiasi messaggio. (Se stessimo lavorando con i reali, il "grafico" di f sarebbe una retta, che quindi è nota quando si conoscono due punti per cui passa.)

ESEMPIO 3.3. Sia $N = 26$, $b = 3$, $a = 3$. Vogliamo spedire la parola CIAO ovvero la stringa numerica 3 9 1 15. Poiché $f(P) = 3P + 3 \pmod{26}$ otteniamo

$$f(3) = 12, \quad f(9) = 30 \equiv 4, \quad f(1) = 6, \quad f(15) = 48 \equiv 22$$

quindi spediamo 12 4 6 22. Vediamo la fase di decrittatura (decodifica). Poiché $f^{-1}(C) = a^{-1}C - a^{-1}b \pmod{N} = 9C - 9 \cdot 3 \pmod{26} \equiv 9C - 1 \pmod{26}$, otteniamo

$$\begin{aligned} f^{-1}(12) &= 107 \equiv 3, & f^{-1}(4) &= 35 \equiv 9, \\ f^{-1}(6) &= 53 \equiv 1, & f^{-1}(22) &= 197 \equiv 15 \end{aligned}$$

ovvero la stringa corrispondente alla parola CIAO.¹

ESEMPIO 3.4. Prendiamo $\mathcal{P} = \mathcal{C} = \mathbb{Z}/N^k\mathbb{Z}$. In questo caso ogni messaggio elementare è una sequenza di k lettere e \mathcal{P} è l'insieme dei blocchi (sequenze) di k lettere ciascuno. La mappa f è la stessa dell'esempio precedente

$$\begin{aligned} f : \mathbb{Z}/N^k\mathbb{Z} &\rightarrow \mathbb{Z}/N^k\mathbb{Z} \\ f(P) &= aP + b \pmod{N^k} \end{aligned}$$

Anche se questo metodo sembra più efficace del precedente, ha lo stesso livello di sicurezza.

Il problema è il seguente. Abbiamo a che fare con blocchi di k cifre. Ora la k -esima cifra N -aria di un blocco crittato (cioè la meno significativa) dipende solo dalla k -esima cifra del blocco di partenza non crittato (mentre una cifra del blocco crittato precedente all'ultima dipende non solo dalla cifra corrispondente del blocco in chiaro, ma anche da quelle che la seguono). La mappa che ne dà la

¹In realtà, per il fatto di usare moduli composti, quali $N = 26$, può accadere che la conoscenza della crittatura di due singole lettere non sia sufficiente ad individuare a e b in modo unico. Il problema è che la matrice del sistema che intendiamo risolvere, che sicuramente ha determinante non multiplo di N , potrebbe non avere determinante primo con N , e quindi non essere invertibile modulo N . Si veda [Kob94, pag. 58] per un esempio.

dipendenza è la seguente:

$$\begin{aligned}\bar{f} : \mathbb{Z}/N\mathbb{Z} &\rightarrow \mathbb{Z}/N\mathbb{Z} \\ \bar{f}(P) &= aP + b \pmod{N}\end{aligned}$$

Che cosa può fare un intruso? Prende la k -esima cifra di un blocco e su quella fa l'analisi di frequenza, quindi determina quale è la regola f sull'ultima cifra di ogni blocco. In altre parole, l'intruso scopre i resti di a e b modulo N , diciamoli a_0 e b_0 , con $0 \leq a_0, b_0 < N$, cioè le cifre delle unità di a e b in base N . Non è poi difficile vedere come l'intruso possa, con ulteriori analisi di frequenza, scoprire una ad una anche le altre cifre di a e b in base N .

Dunque la crittografia a mappe affini su $\mathbb{Z}/N^k\mathbb{Z}$ non è affatto intrinsecamente più sicura della crittografia a mappe affini su $\mathbb{Z}/N\mathbb{Z}$: naturalmente la chiave è k volte più lunga, e per effettuare l'analisi di frequenza con una certa probabilità di successo l'intruso avrà bisogno di conoscere un messaggio crittato k volte più lungo, ma il passo base che dovrà eseguire sarà ugualmente semplice, solo dovrà eseguire il tutto k volte.

Questo tipo di problema si presenta ogni volta che parti della chiave in un sistema di crittografia possono essere ricavate separatamente, o in sequenza come nel caso specifico. Idealmente vorremmo invece che anche ricavare una piccola porzione della chiave sia difficile.

Non possiamo certo migliorare la situazione iterando la crittatura, magari con chiavi diverse, in quanto componendo mappe affini si ottengono ancora mappe affini.

ESEMPIO 3.5. Prendiamo $N = 10$ e $k = 2$. Quindi avremo $\mathcal{P} = \mathbb{Z}/100\mathbb{Z} = \{0, 1, 2, \dots, 99\}$. Prendiamo $a = 3$ e $b = 5$ per cui $f(P) = 3P + 5 \pmod{100}$. Supponiamo che il messaggio elementare sia 42, quindi $f(42) = 3 \cdot 42 + 5 = 131 \equiv 31 \pmod{100}$. Notiamo che l'ultima cifra 1 si ricava solo dall'ultima cifra 2 del numero di partenza applicando \bar{f} :

$$\begin{aligned}\bar{f}(P) &= 3P + 5 \pmod{10} \\ \bar{f}(2) &= 3 \cdot 2 + 5 \equiv 1 \pmod{10}\end{aligned}$$

ESEMPIO 3.6. Prendiamo $\mathcal{P} = (\mathbb{Z}/N\mathbb{Z})^k$, cioè il prodotto diretto di k copie di $\mathbb{Z}/N\mathbb{Z}$. L'insieme \mathcal{P} è quindi formato da blocchi di k lettere ciascuno, pensati come k -uple di interi modulo N , così come sono. Consideriamo ad esempio le 26 lettere dell'alfabeto, per cui alle lettere a,b,c,... corrispondono rispettivamente i numeri 1, 2, 3, ... Quindi data una sequenza di k lettere, la traduco in una sequenza di k numeri (modulo 26). Ottengo quindi un vettore colonna con k entrate a coefficienti in $\mathbb{Z}/N\mathbb{Z}$. Definiamo la mappa f

$$\begin{aligned}f : (\mathbb{Z}/N\mathbb{Z})^k &\rightarrow (\mathbb{Z}/N\mathbb{Z})^k \\ f(P) &= aP + b\end{aligned}$$

dove

$$b \in (\mathbb{Z}/N\mathbb{Z})^k$$

$$a = \text{matrice} \in M_k(\mathbb{Z}/N\mathbb{Z}), \text{ invertibile}$$

Una tale funzione f è detta una mappa affine su $(\mathbb{Z}/N\mathbb{Z})^k$. Notiamo che a è invertibile sse il suo determinante $\det(a)$ è invertibile in $\mathbb{Z}/N\mathbb{Z}$, per cui in generale non è sufficiente la condizione $\det(a) \neq 0$ (basta pensare all'insieme delle matrici a coefficienti interi, in cui una matrice è invertibile sse ha coefficiente ± 1). Dato $P \in (\mathbb{Z}/N\mathbb{Z})^k$ sia $c \in (\mathbb{Z}/N\mathbb{Z})^k$ tale che $f(P) = aP + b = C$, quindi $a^{-1}(aP + b) = a^{-1}C$ e $P = a^{-1}C - a^{-1}b$. Di conseguenza $f^{-1}(C) = a^{-1}C - a^{-1}b$.

Nel caso particolare in cui $a = I$ la matrice identità si ottiene il *codice di Vigenère*, in cui $f(P) = P + b$ (in sostanza si tratta di mettere le lettere in una ruota e poi girare la ruota, usando k chiavi diverse a rotazione, una per ciascuna lettera (o unità elementare) del messaggio).

Chiaramente il codice di Vigenère ha la stessa sicurezza intrinseca di una semplice traslazione (cioè prendendo $k = 1$), per l'osservazione fatta riguardo al metodo delle mappe affini su $(\mathbb{Z}/N^k\mathbb{Z})$. (Un'osservazione analoga vale se si prende per a una matrice scalare, piuttosto che la matrice identità come nel codice di Vigenère.) Invece se prendiamo per a una "generica" matrice invertibile, il problema sollevato in quell'osservazione non si presenterà, e la sicurezza sarà intrinsecamente aumentata. Infatti, una singola componente del vettore che rappresenta un messaggio elementare crittato dipenderà da tutte le componenti del corrispondente messaggio in chiaro. Un'eventuale analisi di frequenza dovrà essere eseguita sugli interi vettori e non sulle singole componenti, e diventerà presto impraticabile. (La chiave qui è individuata da $k^2 + k$ componenti, e per quanto detto non potrà essere ricavata "un pezzetto alla volta", al contrario che nel metodo precedente.)

3.1.1. Conclusioni. Tutti i sistemi visti finora si dicono *a chiave segreta*, o anche a chiave simmetrica, dove *segreta* (attenzione, non *privata*) indica che chi manda e chi riceve, e solo loro, devono conoscere le funzioni f e f^{-1} , e *simmetrica* indica che ricaviamo facilmente f da f^{-1} e viceversa, in un tempo comparabile con il tempo richiesto dai vari algoritmi (di crittatura e decrittatura).

ESEMPIO 3.7. Rifacciamoci al primo esempio, in cui $\mathcal{P} = \mathbb{Z}/N\mathbb{Z}$ e $f(x) = ax + b \pmod{N}$. Se è nota f , conosciamo a e b . Il tempo necessario per ricavare f^{-1} è quello necessario ad invertire a (algoritmo di Euclide), cioè $O(\log^2 N)$ (paragonabile più o meno al tempo per calcolare $f(x)$ per un un singolo valore di x).

3.2. L'idea della crittografia a chiave pubblica

Nel capitolo precedente abbiamo visto *sistemi a chiave segreta*. Un sistema si dirà *sistema a chiave pubblica* (o equivalentemente *privata*) se ricavare f^{-1} dalla conoscenza della funzione f richiede un tempo molto maggiore (ad esempio non polinomiale) rispetto agli algoritmi di codifica e decodifica. In questo caso la chiave di crittatura f può tranquillamente essere resa pubblica, mantenendo privata la

chiave di decrittatura f^{-1} . Quindi

$$\begin{cases} f & \text{è la chiave pubblica: la possono conoscere tutti} \\ f^{-1} & \text{è la chiave privata: la conosce solo la persona che riceve i messaggi} \end{cases}$$

(Notate che *privata* significa nota solo ad una persona, in questo caso il legittimo destinatario dei messaggi, mentre *segreta* significa condivisa da almeno due persone, il mittente ed il destinatario.)

Uno dei vantaggi importanti dei sistemi a chiave pubblica è che non è necessario che le due persone che comunicano si incontrino (o abbiano una comunicazione *sicura*) almeno una volta per accordarsi sulle chiavi.

In breve l'idea è questa:

- Y vuole mandare un messaggio a X
- X conosce f e f^{-1}
- X comunica f pubblicamente a Y
- conoscendo f , Y manda un messaggio a X
- X decodifica il messaggio usando f^{-1} (mentre chiunque intercetti il messaggio non può farlo perché nessun'altro conosce f^{-1} , né può ricavarla facilmente conoscendo f)

Per implementare un sistema di crittografia a chiave pubblica c'è bisogno di una *funzione a senso unico* f (una *one-way function* in inglese): f deve essere iniettiva, e ricavare a da $f(a)$ deve essere molto più difficile che non calcolare $f(a)$ partendo da a . Un esempio di funzione a senso unico è il logaritmo discreto in un campo finito, di cui parleremo più avanti: fissato un generatore g del campo finito \mathbb{F}_q , per ogni intero n è facile (nel senso che si può fare efficientemente) calcolare $g^n \in \mathbb{F}_q$ (ad esempio col metodo dei quadrati ripetuti), mentre è difficile (nel caso specifico, non si sa fare in tempo polinomiale) ricavare n (naturalmente solo modulo $q - 1$, l'ordine di g) da g^n e g .

A volte invece di funzione a senso unico si usa una *trapdoor function*, che è una cosa leggermente diversa (ed infatti l'uso che se ne fa è diverso). In questo caso se si hanno tutte le informazioni a disposizione riguardo ad f e f^{-1} , il calcolo di una o dell'altra possono anche essere dello stesso grado di difficoltà; piuttosto, è difficile calcolare f^{-1} conoscendo solo come si calcola f , quindi è difficile a meno di avere qualche informazione aggiuntiva (la *trapdoor*). Un esempio di trapdoor function è quella del metodo RSA, che vedremo fra poco: qui f^{-1} ha la stessa forma di f , un elevamento a potenza modulare (con un diverso esponente), e quindi è dello stesso ordine di difficoltà, conoscendo gli esponenti; la difficoltà consiste nel ricavare un esponente dalla conoscenza dell'altro.

3.2.1. Firma autenticata. Mettiamoci in questa situazione

Anna	Bruno	
f_A	f_B	chiavi pubbliche
f_A^{-1}	f_B^{-1}	chiavi private

Supponiamo che Anna voglia mandare un messaggio a Bruno.

- Anna spedisce il suo messaggio elementare a Bruno utilizzando la funzione f_B (che Anna conosce perché è la chiave pubblica di Bruno e quindi di dominio pubblico).
- Bruno decodifica il messaggio di Anna usando f_B^{-1} che solo lui conosce.

Ora supponiamo che Anna voglia “firmare” il messaggio prima di spedirlo. Che cos'è una firma? Usando la scrittura su carta si tratterebbe di uno scarabocchio che garantisce al destinatario l'identità di chi lo ha prodotto. Dunque sarà qualcosa che solo Anna deve essere in grado di produrre, e d'altra parte Bruno deve essere in grado di convincersene. Usando la crittografia a chiave pubblica, si fa nel modo seguente.

- Anna per aggiungere in fondo al messaggio la sua firma procede in questo modo. Sceglie un'espressione F da usare come firma *in chiaro*. Idealmente dovrebbe essere qualcosa che la identifichi, quindi non necessariamente segreto, ad esempio qualcosa che contenga il suo nome e cognome, ma magari anche che non sia sempre lo stesso per ogni messaggio, ad esempio che contenga qualche informazione dipendente da esso, o la data in cui viene spedito. Anna calcola $f_A^{-1}(F)$ (cioè una cosa che solo lei può calcolare), e al risultato applica la chiave pubblica di Bruno ottenendo $f_B(f_A^{-1}(F))$, che costituirà la sua firma.
- Una volta ricevuto il messaggio, Bruno per verificare l'autenticità della firma applica alla firma f_B^{-1} , ottenendo $f_B^{-1}(f_B(f_A^{-1}(F))) = f_A^{-1}(F)$. A questo punto applica la chiave pubblica di Anna f_A , che gli permette di ritrovare F . (Notiamo che anche in questo caso Bruno è l'unico che può farlo perché è l'unico a conoscere f_B^{-1}).

Dunque, applicando f_B Anna scrive un messaggio che solo Bruno può decodificare, e applicando f_A^{-1} Anna utilizza una cosa che solo lei conosce e quindi autentica la sua firma.

La firma nella crittografia a chiave pubblica è particolarmente importante, e serve ad esempio ad evitare il rischio seguente (impersonificazione). Il cattivo Carlo (chiedo scusa ai lettori di nome Carlo) rivolgersi ad Anna pretendendo di essere Bruno, comunicandole la sua chiave pubblica (di crittatura) f_C e spacciandola per quella di Bruno. (Qui stiamo assumendo che gli utenti si comunichino a distanza le loro chiavi pubbliche quando ne hanno bisogno, e questa è la sorgente del rischio. Questo tipo di rischio non c'è se le chiavi pubbliche vengono scambiate in modo sicuro, o vengono rese pubbliche e certificate da un organismo fidato.) Quindi Anna, credendo di scrivere a Bruno, potrebbe rispondere a Carlo crittando con f_C , magari rivelandogli cose riservate. Oppure, ancora peggio, dopo aver decrittato il messaggio di Anna con f_C^{-1} , Carlo può modificarlo e spedirlo a Bruno, crittandolo con f_B , spacciandosi quindi per Anna.

3.2.2. Crittografia a chiave pubblica RSA. È uno dei sistemi di crittografia a chiave pubblica più noti e prende il nome da Rivest, Shamir e Adleman (1978).²

²In realtà tale sistema era già stato inventato qualche anno prima da C.L. Cocks (1973), che aveva dovuto tenerlo segreto, lavorando per il governo britannico.

Vediamo in che modo Anna si costruisce la sua chiave pubblica e privata.

- Anna sceglie due primi dispari distinti p_A e q_A molto grandi. Una certa sicurezza è garantita (al giorno d'oggi) prendendo p_A e q_A di circa 100 cifre decimali, ovvero circa 330 binarie. Infatti esistono algoritmi efficienti per trovare primi grandi di queste dimensioni (ed anche molto più grandi), mentre fattorizzare un numero di 200 cifre decimali è ancora fuori portata degli algoritmi e potenza di calcolo oggi disponibili.
- Calcola $n_A = p_A \cdot q_A$ e $\varphi(n_A) = \varphi(p_A) \cdot \varphi(q_A)$.
- Sceglie un intero e_A in modo tale che $(e_A, \varphi(n_A)) = 1$.
- Calcola un inverso $d_A = e_A^{-1} \pmod{\varphi(n_A)}$, cioè trova un intero d_A tale che $e_A d_A \equiv 1 \pmod{\varphi(n_A)}$.

La chiave pubblica (di crittatura) di Anna è data dalla coppia (n_A, e_A) , mentre la sua chiave privata (di decrittatura) è data dalla coppia (n_A, d_A) . Naturalmente n_A è noto a tutti, facendo anche parte della chiave pubblica, mentre la parte veramente privata è d_A . Questa è estremamente difficile da ricavare conoscendo solo la chiave pubblica (n_A, e_A) : precisamente, è dello stesso grado di difficoltà che fattorizzare n . (Giustificeremo presto questa osservazione.)

Ricapitolando, ci troviamo nella seguente situazione

informazioni pubbliche	informazioni private
n_A, e_A	p_A, q_A, d_A

In realtà Anna può benissimo dimenticarsi dei fattori primi p_A e q_A di n_A , che le sono serviti per calcolare d_A ma che non intervengono negli algoritmi di crittatura e decrittatura. (In realtà la conoscenza di p_A e q_A velocizzerebbe certe operazioni grazie al teorema cinese dei resti, ma vedremo in un'osservazione che ciò può esporre a dei rischi.)

Vediamo ora in che modo Bruno manda un messaggio a Anna.

- Sia P l'unità elementare di messaggio che Bruno vuole mandare ad Anna; esso dovrà essere un elemento di $\mathbb{Z}/n_A\mathbb{Z}$, ma possiamo anche identificarlo con un intero non negativo minore di n_A ;
- Bruno, che come tutti conosce la chiave pubblica (n_A, e_A) di Anna, calcola $C \equiv P^{e_A} \pmod{n_A}$;
- Bruno spedisce C ad Anna.

Anna decodifica il messaggio di Bruno nel seguente modo.

- Anna riceve C ;
- Anna, che è l'unica persona che conosce d_A calcola $C^{d_A} = P^{e_A d_A} \equiv P \pmod{n_A}$ (per la dimostrazione di questa equivalenza si veda sotto).

Quindi la funzione di crittatura consiste nell'elevamento all'esponente e_A modulo n_A , mentre la funzione di decrittatura consiste nell'elevamento all'esponente d_A modulo n_A . (Equivalentemente, possiamo dire che $\mathcal{P} = \mathcal{C} = \mathbb{Z}/n_A\mathbb{Z}$, e che le funzioni di crittatura e decrittatura sono l'elevamento ad esponente e_A e d_A nell'anello $\mathbb{Z}/n_A\mathbb{Z}$.)

Dimostriamo ora che $P^{e_A d_A} \equiv P \pmod{n_A}$. Notiamo innanzitutto che $e_A d_A \equiv 1 \pmod{\varphi(n_A)}$, ovvero $e_A d_A = 1 + t\varphi(n_A)$ per qualche intero t . Se sapessimo che

$(P, n_A) = 1$ allora, per il Teorema di Eulero-Fermat otterremo

$$P^{e_A d_A} = P^{1+t\varphi(n_A)} = P \cdot P^{t\varphi(n_A)} \equiv P \pmod{n_A}$$

Vediamo cosa succede nel caso in cui $(P, n_A) \neq 1$.

Ricordo che il Teorema di Eulero-Fermat, cioè $a^{\varphi(n)} \equiv 1 \pmod{n}$, nel caso particolare in cui n è primo continua a valere senza l'ipotesi che $(a, n) = 1$, purché però aumentiamo di uno gli esponenti di entrambi i membri. In effetti questa è la versione piú antica del teorema, che va sotto il nome di

TEOREMA (Piccolo Teorema di Fermat). *Sia p un primo. Per ogni intero a vale $a^p \equiv a \pmod{p}$.*

Una simile estensione vale piú in generale nell'ipotesi che n sia libero da quadrati.

TEOREMA (Estensione del Teorema di Eulero-Fermat). *Sia n un intero libero da quadrati (cioè prodotto di primi distinti), e sia $a \in \mathbb{Z}$ un intero qualunque. Allora*

$$a^{\varphi(n)+1} \equiv a \pmod{n}.$$

DIMOSTRAZIONE. Per il momento evitiamo di supporre che n sia libero da quadrati, e vediamo fin dove possiamo arrivare senza quella ipotesi. Se p è un divisore primo di n , vale $a^{t(p-1)+1} \equiv a \pmod{p}$ per ogni intero a ed ogni intero $t \geq 0$. Lo possiamo mostrare per induzione su t usando il Piccolo teorema di Fermat ($a^{t(p-1)+1} = a^{(t-1)(p-1)} \cdot a^p \equiv a^{(t-1)(p-1)} \cdot a \pmod{p}$), oppure direttamente come nella dimostrazione stessa del Piccolo teorema di Fermat, distinguendo i casi $p \mid a$ e $p \nmid a$. Essendo $\varphi(n)$ un multiplo di $p-1$, avremo dunque $a^{\varphi(n)+1} \equiv a \pmod{p}$, per ogni divisore primo p di n . Concludiamo che $a^{\varphi(n)+1} \equiv a \pmod{m}$, dove m è il prodotto dei divisori primi di n (ciascuno contato una volta sola). Se poi n è libero da quadrati, sarà $m = n$. \square

L'ipotesi che n non sia libero da quadrati non si può togliere, ad esempio se $n = p^2$ (o, piú in generale, se $p^2 \mid n$) basta prendere $a = p$ per avere $p^{\varphi(n)} \equiv 0 \not\equiv p \pmod{n}$. (L'ostacolo è proprio che $\mathbb{Z}/p^2\mathbb{Z}$ contiene elementi nilpotenti, quali $p + p^2\mathbb{Z}$.)

Ricapitolando, la funzione f_A di Anna è

$$f_A(P) = P^{e_A} \pmod{n_A}$$

e la sua inversa f_A^{-1} è

$$f_A^{-1}(C) = C^{d_A} \pmod{n_A}$$

OSSERVAZIONI. (1) In tutto il ragionamento (e in particolare nell'enunciato della generalizzazione del Teorema di Eulero-Fermat) possiamo usare la funzione λ di Carmichael al posto della funzione φ di Eulero, cioè rimpiazzare $\varphi(n)$ con $\lambda(n) = [p-1, q-1] = pq/(p-1, q-1)$. In particolare, la chiave privata d_A di Anna dovrà essere soltanto un inverso di e_A modulo $\lambda(n)$, piuttosto che modulo $\varphi(n)$. In questo modo si possono semplificare leggermente i calcoli, e diminuire la lunghezza delle chiavi.

(Solo “leggermente”, in quanto se $(p-1, q-1)$ è eccessivamente grande il metodo perde gran parte della sua sicurezza, come vedremo piú avanti.)

- (2) Negli esempi di crittografia a chiave simmetrica avevamo sempre preso $\mathcal{P} = \mathcal{C}$. Ora non possiamo piú. In questo caso stiamo considerando

$$\mathcal{P} = \{\text{interi } P : 0 \leq P < N^k\}, \quad \text{con } N^k \leq n_A.$$

Non possiamo però prendere $\mathcal{C} = \mathcal{P}$ perché calcolando $P^{e_A} \pmod{n_A}$ è possibile che usciamo dall’insieme \mathcal{P} . Dobbiamo prendere

$$\mathcal{C} = \{\text{interi } C : 0 \leq C < N^l\}, \quad \text{con } N^l \geq n_A.$$

I blocchi di messaggio in chiaro saranno lunghi k , mentre i blocchi di messaggio crittato saranno di lunghezza l , maggiore di k . Per la scelta dei parametri procediamo nel modo seguente. Fissiamo prima N^k e N^l di circa 200 cifre e con una certa distanza. Quindi scegliamo p_A e q_A in modo che n_A cada in questo intervallo.

- (3) Quella che abbiamo dato è solo una descrizione sommaria del metodo RSA. In una implementazione pratica bisogna evitare molte situazioni che ne minerebbero la sicurezza. In particolare, i primi p e q scelti devono soddisfare ad alcuni requisiti, i piú importanti dei quali sono: i due primi non dovrebbero essere troppo vicini fra loro (ad esempio, uno dovrebbe essere alcune cifre decimali piú lungo dell’altro), $p-1$ e $q-1$ dovrebbero avere un massimo comun divisore piuttosto piccolo, e ciascuno dovrebbe avere almeno un divisore primo grande. Infatti, in queste situazioni l’intero n diventa piú facile da fattorizzare che in generale: ad esempio, se p e q sono troppo vicini, n si fattorizza facilmente con il metodo di fattorizzazione di Fermat (cercare per tentativi un intero a relativamente poco maggiore di $\lceil \sqrt{n} \rceil$ tale che $a^2 - n$ sia un quadrato perfetto, diciamo b^2 , dopodiché $n = (a-b)(a+b)$). Vedremo la ragione della seconda restrizione nella Sezione 4.3. Infine, se $p-1$ (o $q-1$) è prodotto di primi “piccoli” (cioè, come si dice, è *smooth*), allora n diviene suscettibile ad essere fattorizzato mediante il metodo $p-1$ di Pollard.

Anche nella scelta di e ed f bisogna stare attenti. In particolare, nessuno dei due deve essere troppo piccolo (come si potrebbe essere tentati dall’esigenza di velocizzare la crittatura o la decrittatura), altrimenti violare il metodo RSA diventa sostanzialmente piú facile che nel caso generale. (Questo non è ovvio.)

- (4) Ci sono tanti altri trabocchetti in cui non bisogna cadere. Ne illustriamo brevemente solo uno.

Vi sono applicazioni in cui la decrittatura (in realtà, nella forma di “porre una firma elettronica”) è eseguita da una *smartcard* (una tessera contenente un microprocessore), che autentica la sua identità (o meglio quella di chi la possiede) restituendo “firmato” un messaggio ricevuto. Chi se ne impossessi illegalmente, anche temporaneamente, potrebbe “clonarla” se riesce a scoprirne la chiave privata d . Per facilitare l’operazione di elevamento all’esponente e modulo $n = pq$ da parte della smartcard, di memoria e potenza di calcolo limitata, si potrebbe farle

eeguire la potenza modulo p e modulo q (ciascuno in un ottavo del tempo necessario alla potenza modulo n , data la stima $O(\log^3 n)$), e quindi ottenerne il risultato modulo n tramite il Teorema cinese dei resti (in un tempo relativamente trascurabile, anche perché la smartcard non deve eseguire l'algoritmo di Euclide, basta che ne abbia in memoria il risultato, degli interi u e v tali che $1 = up + vq$; dunque il tempo complessivo dunque il tempo complessivo diventa circa un quarto di quanto servirebbe altrimenti).

Supponiamo ora che un malintenzionato, nel tempo in cui è in possesso della smartcard, possa farle “firmare” un alto numero di volte un qualsiasi messaggio M da lui scelto. È sufficiente che anche una sola volta si verifichi nella smartcard un solo errore hardware (che potrebbe anche essere in qualche modo “forzato” dal delinquente) nel calcolo di una potenza, diciamo quella modulo p (ma non di quella modulo q), per permettere al malintenzionato di fattorizzare n , nel modo seguente: se $S \equiv M^d \pmod{n}$ è la firma corretta ed \tilde{S} è quella errata, S ed \tilde{S} saranno congrui modulo q ma non modulo p , da cui $(\tilde{S} - S, n) = p$.

3.2.3. Firma autenticata nel metodo RSA. Nella nostra presentazione iniziale della firma autenticata abbiamo preso implicitamente $\mathcal{P} = \mathcal{C}$. Come abbiamo visto poco fa, utilizzando il metodo RSA dobbiamo rivedere qualcosa.

Quando Anna spedisce la firma a Bruno dobbiamo distinguere due casi. Anzitutto Anna sceglie come sua “firma in chiaro” un intero F minore sia di n_A che di n_B .

- (1) Se $n_A < n_B$ procediamo come descritto in precedenza. Quindi per firmare Anna calcola $F^{d_A} \pmod{n_A}$, eleva il risultato ad e_B e riduce modulo n_B .
- (2) Se $n_A > n_B$, non va bene che Anna firmi con $(F^{d_A} \pmod{n_A})^{e_B} \pmod{n_B}$ come nel caso precedente. Infatti $F^{d_A} \pmod{n_A}$ è un intero minore di n_A , e la seconda funzione che calcoliamo, l'elevamento ad esponente $e_B \pmod{n_B}$, è iniettiva sugli interi minori di n_B , ma non necessariamente sugli interi minori di n_A . Risolviamo il problema scambiando l'ordine con cui vengono applicate le due funzioni (e questo anche Bruno lo sa, in quanto sia n_A che n_B sono pubblici). Quindi per firmare Anna calcola $F^{e_B} \pmod{n_B}$, e poi eleva il risultato a d_A riducendolo modulo n_A .

3.2.4. Scoprire la chiave privata d è tanto difficile quanto fattorizzare $n = pq$. Abbiamo detto che la conoscenza di $\varphi(n)$, dove $n = p \cdot q$, equivale alla conoscenza della fattorizzazione di n . Supponiamo che una terza persona scopra una informazione più debole di $\varphi(n)$, scopra un multiplo (non troppo grande) di $\varphi(n)$, o anche di $[p - 1, q - 1]$. Faremo vedere che da questa conoscenza si riesce a fattorizzare n .

Supponiamo che un intruso riesca a trovare un intero d tale che $a^{ed} \equiv a \pmod{n}$ per ogni intero a (o, equivalentemente, per ogni intero a primo con n , grazie all'estensione del Teorema di Eulero-Fermat). Ricordo che e è la chiave pubblica, cioè un intero che Anna ha scelto primo con $\varphi(n)$, e d è un inverso di e modulo $\varphi(n)$, anzi, basta modulo $[p - 1, q - 1]$, che solo Anna dovrebbe conoscere.

Dalla nostra ipotesi segue che $ed - 1$ è un multiplo dell'esponente di $U(\mathbb{Z}/n\mathbb{Z})$, che sappiamo essere $\lambda(n) = [p - 1, q - 1]$. Quindi l'intruso si è impadronito di un multiplo non troppo grande m di $[p - 1, q - 1]$, diciamo minore di n^2 .³ Ora mostreremo come è possibile usare m per fattorizzare n . La supposta difficoltà di fattorizzare n implica dunque che dovrebbe essere molto difficile per un intruso scoprire d .

Il punto fondamentale è che la classe resto di 1 ha esattamente quattro radici quadrate distinte in $\mathbb{Z}/n\mathbb{Z}$. Infatti per il Teorema cinese dei resti

$$\mathbb{Z}/n\mathbb{Z} \cong \mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z}$$

Poichè $\mathbb{Z}/p\mathbb{Z}$ e $\mathbb{Z}/q\mathbb{Z}$ sono campi e quindi contengono esattamente due radici quadrate di 1 distinte, segue che le radici di 1 in $\mathbb{Z}/n\mathbb{Z}$ corrispondono alle coppie $(\pm 1, \pm 1)$ di $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z}$. Nella corrispondenza abbiamo

$$\begin{array}{ccc} \mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z} & \rightarrow & \mathbb{Z}/n\mathbb{Z} \\ (1, 1) & \mapsto & 1 \\ (-1, -1) & \mapsto & -1 \\ (1, -1) & \mapsto & x_0 \\ (-1, 1) & \mapsto & -x_0 \end{array}$$

dove $x_0 \neq \pm 1$. Non appena conosco una radice quadrata di 1 diversa da quelle ovvie, 1 e -1 , diciamo x_0 , riesco a fattorizzare n , infatti da $x_0^2 \equiv 1 \pmod{pq}$ segue che $(x_0 - 1)(x_0 + 1) \equiv 0 \pmod{pq}$, quindi $pq \mid (x_0 - 1)(x_0 + 1)$, e poichè sia $x_0 - 1$ che $x_0 + 1$ sono non nulli deve essere $p \mid x_0 - 1$ e $q \mid x_0 + 1$ o viceversa. Utilizzando l'algoritmo di Euclide possiamo quindi calcolare

$$p = (x_0 - 1, pq), \quad q = (x_0 + 1, pq).$$

Il problema è quindi trovare una radice quadrata dell'unità non banale. Riprendiamo quindi dalla conoscenza di m tale che $a^m \equiv 1 \pmod{n}$ per ogni a con $(a, n) = 1$.

Cominciamo con l'osservare che m è pari: lo si vede prendendo $a = -1$ (o anche perché $p - 1 \mid m$). Verifichiamo ora se vale $a^{m/2} \equiv 1 \pmod{n}$ per ogni a con $(a, n) = 1$. Problema: come facciamo a verificarlo per ogni a ? Lo facciamo in *modo probabilistico*, testando (qualche decina di) a diversi a caso. Infatti se vale $a^{m/2} \not\equiv 1 \pmod{n}$ per almeno un a , allora questo varrà per almeno il 50% degli a tali che $(a, n) = 1$. Quindi facendo ad esempio 30 tentativi abbiamo al massimo $1/2^{30}$ possibilità di sbagliarci. A ogni scelta casuale di a abbiamo almeno

³La chiave privata d è unicamente determinata solo se richiediamo che $0 < d < [p - 1, q - 1]$ e probabilmente sarà scelta da Anna con questa condizione, per efficienza. Se anche e soddisfa un'analogha condizione, come è lecito e conveniente richiedere, $ed - 1$ sarà minore di $[p - 1, q - 1]^2$ e quindi minore di $n^2/4$, visto che $(p - 1, q - 1) \geq 2$. In generale questo è tutto quanto si può dire della grandezza di $m = ed - 1$, nel senso che in generale esso potrà anche essere dell'ordine di grandezza di $n^2/4$. Essendo $[p - 1, q - 1]$ al più $n/2$ si capisce quanto la conoscenza del multiplo $ed - 1$ di $[p - 1, q - 1]$ sia, almeno in apparenza, molto meno che conoscere $[p - 1, q - 1]$ stesso (o $\varphi(n)$). Come stiamo per vedere, ciò è solo apparenza.

probabilità $1/2$ di trovare un a tale che $a^{m/2} \not\equiv 1 \pmod{n}$ (supposto che un tale a esista).⁴

Se $a^{m/2} \equiv 1 \pmod{n}$ risulta vero per ogni intero a tale che $(a, n) = 1$ (o meglio per un numero sufficiente di interi a scelti a caso), dimezziamo $m/2$ e ripetiamo il test controllando se vale $a^{m/4} \equiv 1 \pmod{n}$ per ogni a con $(a, n) = 1$, o meglio, per un certo numero di tali a scelti a caso (notando che ragionando come prima con $m/2$ al posto di m segue che $m/2$ deve essere pari).

Prima o poi troveremo un esponente $k = m/2^i$ tale che

$$\begin{aligned} a^{2k} &\equiv 1 \pmod{n} && \text{per ogni } a \text{ con } (a, n) = 1, \\ a^k &\not\equiv 1 \pmod{n} && \text{per qualche } a \text{ con } (a, n) = 1. \end{aligned}$$

Dunque l'esponente di $U(\mathbb{Z}/n\mathbb{Z})$, vale a dire $[p-1, q-1]$, divide $2k$ ma non k . Abbiamo due possibili casi:

- $p-1$ non divide k e $q-1$ divide k (o viceversa);
- né $p-1$ né $q-1$ dividono k .

(Il secondo avverrà se la massima potenza di 2 che divide $p-1$ e $q-1$ è la stessa, il primo altrimenti.)

Consideriamo il primo caso: $p-1$ non divide k e $q-1$ divide k . Un intero a con $(a, n) = 1$ scelto a caso soddisferà

$$\begin{cases} a^k \equiv 1 \pmod{p} \\ a^k \equiv 1 \pmod{q} \end{cases} \quad \text{oppure} \quad \begin{cases} a^k \equiv -1 \pmod{p} \\ a^k \equiv 1 \pmod{q} \end{cases}$$

ciascuno con probabilità $1/2$. Infatti a^k non può che essere congruo a ± 1 modulo p , in quanto il suo quadrato è congruo a 1 e \mathbb{F}_p è un campo; sarà 1 se e solo se \bar{a} appartiene al nucleo dell'omomorfismo $\mathbb{F}_p^* \rightarrow \mathbb{F}_p^*$ tale che $\bar{a} \mapsto \bar{a}^k$, e quindi in esattamente metà dei casi (perché il nucleo ha indice 2 in \mathbb{F}_p^* , avendo l'immagine ordine 2). Nel secondo caso a^k è una radice quadrata dell'unità non banale e riusciamo a fattorizzare n , infatti $q = (a^k - 1, pq)$. Nel primo caso invece dobbiamo tentare con un altro a .

Consideriamo ora il secondo caso: né $p-1$ né $q-1$ dividono k . Analogamente a prima avremo che

$$\begin{aligned} a^k &\equiv \pm 1 \pmod{p}, && \text{ciascuno per esattamente metà degli } \bar{a} \in U(\mathbb{Z}/p\mathbb{Z}), \\ a^k &\equiv \pm 1 \pmod{q}, && \text{ciascuno per esattamente metà degli } \bar{a} \in U(\mathbb{Z}/q\mathbb{Z}). \end{aligned}$$

Quindi in $U(\mathbb{Z}/n\mathbb{Z}) \cong U(\mathbb{Z}/p\mathbb{Z}) \times U(\mathbb{Z}/q\mathbb{Z})$, interpretando ogni classe resto $\bar{a} \in U(\mathbb{Z}/n\mathbb{Z})$ come coppia, avremo che esattamente

- $1/4$ delle coppie soddisfa $a^k \equiv 1 \pmod{p}$ ed $a^k \equiv 1 \pmod{q}$,

⁴Cerchiamo di spiegare meglio questo fatto. Poiché $(a, n) = 1$, abbiamo che $a \in (\mathbb{Z}/n\mathbb{Z})^*$, il cui ordine è $\varphi(n)$. Consideriamo gli \bar{a} tali che $\bar{a}^{m/2} = \bar{1}$. Essi formano un sottogruppo di $(\mathbb{Z}/n\mathbb{Z})^*$ (formando un sottoinsieme chiuso rispetto a prodotto e inverso). Quindi se non costituisce tutto il gruppo, il suo ordine sarà al massimo la metà dell'ordine del gruppo (per il Teorema di Lagrange): infatti, l'ordine di tale sottogruppo è un divisore di $\varphi(n)$, e se è un divisore proprio sarà $\leq \varphi(n)/2$. (Più precisamente, anche se è superfluo per il nostro ragionamento, sarà $\varphi(n)/2$ o $\varphi(n)/4$ a seconda che la massima potenza di 2 che divide $p-1$ e $q-1$ sia diversa o la stessa.)

- 1/4 delle coppie soddisfa $a^k \equiv 1 \pmod{p}$ ed $a^k \equiv -1 \pmod{q}$,
- 1/4 delle coppie soddisfa $a^k \equiv -1 \pmod{p}$ ed $a^k \equiv 1 \pmod{q}$,
- 1/4 delle coppie soddisfa $a^k \equiv -1 \pmod{p}$ ed $a^k \equiv -1 \pmod{q}$.

Nei due casi centrali a^k è una radice quadrata dell'unità modulo n diversa da ± 1 , quindi riusciamo a fattorizzare n . Negli altri due casi non otteniamo nulla e dobbiamo cambiare a .

- OSSERVAZIONI. (1) Naturalmente l'efficienza del metodo dipende dalla grandezza del multiplo m di $[p-1, q-1]$ che l'intruso ha a disposizione. Non stiamo supponendo che l'intero d trovato dall'intruso sia necessariamente quello tale che $0 < d < [p-1, q-1]$, qualunque altro congruo ad esso modulo $[p-1, q-1]$ farà la stessa funzione (e comunque l'intruso, a differenza di Anna, non sarà in grado di ridurlo modulo $[p-1, q-1]$, in quanto non conosce quest'ultimo). Tuttavia è ragionevole supporre che, in qualunque modo l'intruso abbia scoperto un tale d , esso (e con lui $m = ed - 1$) abbia un ordine di grandezza non esagerato rispetto a quello di n^2 , per cui il metodo descritto per fattorizzare n , pur probabilistico, è efficiente. Notate che se non mettiamo una restrizione alla grandezza di m , è facile trovarne uno, ad esempio un multiplo di $[p-1, q-1]$ è certamente $n!$; questo multiplo però è enorme, e quindi inutilizzabile.
- (2) L'idea di cercare due radici quadrate di un numero modulo n che non siano uguali o opposte modulo n è ricorrente nei metodi di fattorizzazione (di cui quello visto comunque *non* si può considerare un esempio, perché nelle situazioni in cui serve fattorizzare un intero n non si avrà in generale a disposizione un multiplo di $\varphi(n)$), da quello di Fermat a vari dei metodi più moderni.

3.2.5. Testa o croce telefonico. Anna e Bruno vogliono giocare a testa o croce per telefono. Esiste un metodo per farlo basato sul fatto che mentre esiste un algoritmo efficiente per calcolare le radici quadrate modulo un primo p , non ne esiste uno per calcolare le radici quadrate modulo un prodotto $n = pq$ di due primi distinti, a meno di non conoscere la fattorizzazione di n , e quindi poter usare il Teorema cinese dei resti. In altre parole, la funzione *elevamento al quadrato modulo n* è una trapdoor function, e la trapdoor è la conoscenza della fattorizzazione di n . Vediamo come essa viene utilizzata.

Anna sceglie due primi grandi p e q , calcola $n = pq$ e lo manda a Bruno.

Bruno sceglie un numero a tale che $(a, n) = 1$, lo eleva al quadrato e invia a^2 a Anna. Il numero a^2 , essendo un quadrato ha 4 radici quadrate distinte modulo n (che solo Anna può calcolare, conoscendo p e q).

Anna estrae le radici quadrate $\pm x$ di a^2 modulo p , e $\pm y$ modulo q . Con il Teorema cinese dei Resti calcola le 4 radici quadrate $\pm b, \pm a$ di a^2 modulo n (con $a \not\equiv \pm b \pmod{n}$). Delle 4 radici trovate Anna non sa quale è quella scelta inizialmente da Bruno. A questo punto ne sceglie una a caso (cioè "lancia il dado") e la invia a Bruno.

- Se Anna sceglie $\pm b$, Bruno ha vinto e lo può dimostrare fattorizzando n .

Infatti ora Bruno conosce due radici quadrate distinte di a^2 , quindi

$$a^2 \equiv b^2 \pmod{n}$$

$$a^2 - b^2 \equiv 0 \pmod{n}$$

$$(a - b)(a + b) \equiv 0 \pmod{n}$$

Quindi $n \mid (a - b)(a + b)$ (ma n non divide né $a - b$ né $a + b$ perché $a \not\equiv \pm b \pmod{n}$), per cui $p \mid a - b$ e $q \mid a + b$ o viceversa. Infine

$$p = (a - b, n), \quad q = (a + b, n)$$

- Se Anna sceglie $\pm a$, Bruno ha perso in quanto non è in grado di fattorizzare n .

Naturalmente nel secondo caso Anna non può sapere con certezza di aver vinto, ma si presuppone che Bruno *voglia* vincere.

3.3. Il logaritmo discreto

Quando abbiamo discusso il sistema di crittografia RSA, abbiamo visto che era basato sulla facilità di trovare due primi grandi e moltiplicarli tra loro, e sulla difficoltà di fare l'operazione inversa. Ci sono altri procedimenti fondamentali in teoria dei numeri che godono di queste proprietà *a senso unico*.

Sia G un gruppo e $g \in G$. Consideriamo la seguente mappa, la *funzione esponenziale di base g* ,

$$\begin{aligned} \mathbb{Z} &\rightarrow G \\ n &\mapsto g^n \end{aligned}$$

Si tratta di un omomorfismo di gruppi tra $(\mathbb{Z}, +, 0)$ e $(G, \cdot, 1)$. Indichiamo con $|g|$ l'ordine di g in G . Possiamo ottenere un omomorfismo iniettivo $\mathbb{Z}/|g|\mathbb{Z} \rightarrow G$, la cui immagine è quindi invertibile. L'omomorfismo inverso, definito sull'immagine $\langle g \rangle$, è la mappa

$$\log_g : \langle g \rangle \rightarrow \mathbb{Z}/|g|\mathbb{Z}$$

detta *logaritmo discreto in base g* . In altre parole, se $a \in \langle g \rangle$, allora $\log_g a$ sarà quell'esponente y (definito a meno di multipli di $|g|$) tale che $g^y = a$, cioè

$$\begin{aligned} \log_g : \langle g \rangle &\rightarrow \mathbb{Z}/|g|\mathbb{Z} \\ a &\mapsto y + |g|\mathbb{Z} \text{ tale che } g^y = a \end{aligned}$$

In generale è facile calcolare la prima mappa (elevamento a potenza), ma è difficile calcolare la seconda (logaritmo discreto).

OSSERVAZIONI. Vediamo che relazione c'è con il logaritmo definito sui reali. Consideriamo l'isomorfismo di gruppi

$$\begin{aligned} \exp : \mathbb{R} &\rightarrow (\mathbb{R}^+, \cdot) \\ x &\mapsto e^x \\ \log_e : (\mathbb{R}^+, \cdot) &\rightarrow \mathbb{R} \\ y &\mapsto \log_e y \end{aligned}$$

Si tratta di mappe del campo nel campo che legano la struttura additiva e la struttura moltiplicativa del campo \mathbb{R} . (Al posto di \mathbb{R} possiamo usare \mathbb{C} o altri

opportuni campi con caratteristica zero che abbiano certe proprietà, come i numeri p -adici. Le due mappe non vanno bene per un campo di caratteristica p .)

Un caso interessante è quando $G = \mathbb{F}_q^*$ è il gruppo moltiplicativo di un campo finito (dove q è un primo o una potenza di primo).⁵ Se $b \in \mathbb{F}_q^*$ (o meglio ancora è un *generatore*), la mappa

$$\begin{aligned} \mathbb{Z} &\rightarrow \mathbb{F}_q^* \\ n &\mapsto b^n \end{aligned}$$

è facile da calcolare con il metodo dei quadrati ripetuti (in $O(\log^3 q)$ operazioni bit), ma è difficile da invertire, nel senso che non esistono algoritmi veloci per farlo.

Analizziamo adesso tre tipi di sistemi di crittografia che utilizzano questa funzione a senso unico. Non trattandosi di una *trapdoor function* (allo stato attuale delle conoscenze, non esiste alcuna informazione ulteriore, eventualmente da tenere privata, che permetta di calcolare il logaritmo discreto efficientemente in generale), non si potrà costruire ad esempio un analogo del metodo RSA basato sul logaritmo discreto: la funzione a senso unico si utilizza in modi diversi.

3.3.1. Il sistema di Diffie-Hellman. Di solito si usano sistemi di crittografia a chiave segreta per scambiare messaggi (perché sono più veloci), e di tanto in tanto ci si scambia la chiave con un sistema a chiave pubblica.

Supponiamo che Anna e Bruno vogliano concordare una chiave a distanza (il più possibile casuale) da utilizzare poi in un sistema di crittatura a chiave segreta. Quindi *pubblicamente*

- fissano una corrispondenza dell'insieme delle chiavi possibili con gli elementi non nulli di un grande campo finito \mathbb{F}_q (con q primo o, più in generale, una potenza di un primo);
- fissano un generatore g di \mathbb{F}_q^* (o comunque un elemento di ordine grande se è difficile trovare un generatore; in questo secondo caso le chiavi che si potranno ottenere saranno solo le potenze di g);
- Anna sceglie un intero a e comunica g^a ;
- Bruno sceglie un intero b e comunica g^b .

A questo punto sia Anna che Bruno (e solo loro) sanno calcolare $g^{ab} = (g^a)^b = (g^b)^a$.

Ci troviamo quindi nella seguente situazione

informazioni pubbliche	informazioni private
g, g^a, g^b	a, b, g^{ab}

Il punto è che è difficile calcolare g^{ab} partendo da g^a e g^b . Tale problema è detto *problema di Diffie-Hellman*. Si pensa che sia della stessa difficoltà di calcolare a e

⁵Nei metodi basati sul logaritmo discreto qui descritti, il gruppo \mathbb{F}_q^* si può sostituire con altri per cui il problema del logaritmo discreto sia considerato difficile, ad esempio il gruppo dei punti razionali di una curva ellittica su un campo finito. Sarebbe invece una pessima idea utilizzare il gruppo $U(\mathbb{Z}/p^\alpha\mathbb{Z})$: si vede facilmente che esso non offrirebbe maggiore sicurezza che $U(\mathbb{Z}/p\mathbb{Z}) = \mathbb{F}_p^*$.

b a partire da g^a e g^b , ovvero di risolvere il problema del logaritmo discreto (anche se il solo verso dell'equivalenza che si sa dimostrare è quello ovvio).

ESERCIZIO 3.8. Mostrate che la seguente è una forma essenzialmente equivalente del problema di Diffie-Hellman: conoscendo tre qualsiasi dei quattro elementi g, g^a, g^b, g^{ab} di un campo finito (noto) \mathbb{F}_q , calcolare il quarto. (Cioè mostrate che non importa quale dei quattro non si conosce, il problema è essenzialmente lo stesso; potete assumere che sia a che b siano primi con $|g|$, il che sarà altamente probabile se a e b sono random.)

3.3.2. Il sistema di Massey-Omura. Questo sistema non è propriamente a chiave pubblica, perché non c'è una chiave pubblica, ma soltanto chiavi private. Per un confronto possiamo interpretare il sistema RSA nel modo seguente. Anna possiede una cassetta con un lucchetto, chiunque può infilarci un messaggio e chiudere il lucchetto (il che corrisponde a crittare con la chiave pubblica di Anna), e nessuno potrà riaprirlo tranne Anna che possiede la chiave (privata). Chiaramente per scambiare messaggi fra Anna e Bruno servono due cassette, una per ciascuno, ognuna con un suo lucchetto.

Nel sistema di Massey-Omura c'è una sola cassetta, con due serrature affiancate. Ciascuna si apre o chiude soltanto con la chiave corrispondente, una delle quali è in possesso di Anna, l'altra di Bruno. La cassetta è si apre soltanto quando entrambe le serrature sono aperte. Vediamo come funziona in pratica. Viene fissato pubblicamente un campo \mathbb{F}_q .

- Anna sceglie un intero e_A primo con $q-1$ (l'esponente di crittatura, ovvero la chiave per chiudere);
- Anna calcola (con l'algoritmo di Euclide) un inverso d_A di e_A modulo $q-1$ (esponente di decrittatura, ovvero la chiave per aprire);
- analogamente Bruno calcola e_B e d_B inversi uno dell'altro modulo $q-1$.

In questo sistema, a differenza del metodo RSA, sia d_A che e_A restano privati (e così d_B ed e_B). Notate infatti che conoscendone uno è facile per chiunque ricavare l'altro, calcolandone l'inverso modulo $q-1$.

- Anna traduce il suo messaggio elementare in un elemento $P \in \mathbb{F}_q$ e spedisce a Bruno P^{e_A} (Anna chiude la serratura);
- Bruno calcola $(P^{e_A})^{e_B} = P^{e_A e_B}$ e lo rispedisce a Anna (Bruno chiude);
- Anna calcola $(P^{e_A e_B})^{d_A} = P^{e_B}$ e lo rispedisce a Bruno (Anna apre);
- Bruno calcola $(P^{e_B})^{d_B} = P$ e scopre il messaggio (Bruno apre).

Ci sono quindi tre passaggi

$$A \rightarrow B \rightarrow A \rightarrow B$$

e in ogni passaggio almeno una delle due serrature è sempre chiusa. Il sistema non usa alcuna funzione a senso unico, ed infatti le funzioni usate sono funzioni *elevamento a potenza* piuttosto che *esponenziali* e relativi logaritmi. ⁶

⁶Attenzione, nel metodo RSA si usano funzioni elevamento a potenza che sono a senso unico, anzi con trapdoor, perché si fanno modulo un intero che non è primo; qui invece si fanno modulo un primo o, più in generale, in un campo finito. Parliamo qui del metodo di Massey-Omura perché la sua sicurezza si basa sulla presunta difficoltà del problema di Diffie-Hellmann, che è

Al posto dell'elevamento a potenza in un campo finito si possono usare altre funzioni, ma notate anche che è essenziale che le operazioni di crittatura e decrittatura dei due utenti commutino, cioè si possano fare in un ordine qualsiasi dando lo stesso risultato.

OSSERVAZIONI. È fondamentale associare a questo sistema di firma autenticata. Infatti se una terza persona C individua il messaggio p^{e_A} inviato da Anna, può rispondere con $(p^{e_A})^{e_C}$, spacciandosi per Bruno, e dall'ingenua risposta $(p^{e_A e_C})^{d_A} = p^{e_C}$ di Anna ricavare e leggere il messaggio inteso per Bruno. (Cosa che non può succedere con il metodo RSA.)

3.3.3. Il sistema di El-Gamal. Con questo metodo si fissano pubblicamente un campo finito \mathbb{F}_q e un generatore g di \mathbb{F}_q^* .

Supponiamo che Bruno voglia mandare un messaggio P a Anna.

- Anna sceglie un intero $1 < a < q - 1$ (la sua chiave privata), e rende pubblico g^a (la sua chiave pubblica);
- Bruno sceglie un intero $1 < k < q - 1$ e spedisce a Anna la coppia $(g^k, P g^{ak})$ (Bruno conosce g^a e g che sono pubblici);
- Anna calcola $(g^k)^a = g^{ak}$ (cosa che solo lei può fare dato che è l'unica a conoscere a);
- Anna ricava l'inverso di g^{ak} nel campo \mathbb{F}_q (cosa che è facile da calcolare), e calcola $P g^{ak} (g^{ak})^{-1} = P$.

Una persona che intercetta il messaggio dovrebbe ricavare g^{ak} conoscendo g, g^a, g^k , dovrebbe cioè affrontare il problema di Diffie-Hellman. Un possibile vantaggio di tale metodo è che k può essere cambiato per ogni messaggio.

3.3.4. L'algoritmo di Silver, Pohlig e Hellman. Affinché i metodi basati sul logaritmo discreto siano sicuri è necessario che $q - 1$ non sia prodotto di fattori primi piccoli, altrimenti il logaritmo discreto si calcola in modo efficiente. Per mostrarlo, iniziamo studiando un esempio estremo, in cui $q = p$ è un primo di Fermat, e quindi $q - 1$ è una potenza di 2.

ESEMPIO 3.9. Si verifica facilmente che 5 è una radice primitiva modulo il primo di Fermat $F_3 = 2^{2^3} + 1 = 257$. (Lo è anzi modulo ogni primo di Fermat diverso da 5.) Calcoliamo il logaritmo discreto di 7 in base 5 nel campo \mathbb{F}_{257} . Cerchiamo quindi l'unico intero $0 \leq x < 256$ tale che $5^x \equiv 7 \pmod{257}$. Scrivendo l'esponente x in base 2, ne calcoleremo in successione le cifre binarie $x_0, x_1, \dots, x_7 \in \{0, 1\}$.

Infatti, elevando $5^{x_0 + 2x_1 + 4x_2 + \dots + 2^7 x_7} \equiv 7 \pmod{257}$ all'esponente $2^7 = 128$ otteniamo che $5^{2^7 x_0} \equiv 7^{2^7} \equiv -1 \pmod{257}$, da cui $x_0 = 1$, in quanto $5^{2^7} \equiv -1 \pmod{257}$. Ma allora $5^{2x_1 + 2^2 x_2 + \dots + 2^7 x_7} \equiv 7 \cdot 5^{-x_0} \equiv 7/5 \equiv 207 \equiv -50 \pmod{257}$. Il passo successivo è calcolare x_1 , e si può pensare come analogo al passo precedente rimpiazzando 5 con $5^2 = 25$ (e il secondo membro 7 con -50 , ovviamente).

collegato al logaritmo discreto. Notate anche che pur non essendo un metodo a chiave pubblica, non si può nemmeno dire a chiave segreta: le chiavi sono private, quindi non serve che i due partecipanti condividano alcun segreto.

Significa dunque calcolare $5^{2^6 \cdot 2x_1} \equiv (-50)^{2^6} \equiv -1 \pmod{257}$, da cui deduciamo che $x_1 = 1$. Ma allora $5^{4x_2 + \dots + 2^7 x_7} \equiv 7 \cdot 5^{-x_0 - 2x_1} \equiv -50/5^2 \equiv -2 \pmod{257}$. Il passo successivo è $5^{2^5 \cdot 2^2 x_2} \equiv (-2)^{2^5} \equiv 1 \pmod{257}$, da cui $x_2 = 0$, e quindi $5^{8x_3 + \dots + 2^7 x_7} \equiv 7 \cdot 5^{-x_0 - 2x_1 - 4x_2} \equiv -2 \pmod{257}$ senza ulteriori calcoli. Così nei prossimi due passi troviamo che $5^{2^4 \cdot 2^3 x_3} \equiv (-2)^{2^4} \equiv 1 \pmod{257}$, da cui $x_3 = 0$, e $5^{2^3 \cdot 2^4 x_4} \equiv (-2)^{2^3} \equiv -1 \pmod{257}$, da cui $x_4 = 1$. (Gli ultimi tre passaggi non ci sono costati alcun calcolo, grazie al fatto fortuito che $(-2)^{2^3} = 257 - 1$; anche a parte questa particolarità, è chiaro che sequenze di bit nulli fanno risparmiare calcoli.) Riassumendo i calcoli fatti fin qui, e continuandoli fino alla fine, troviamo:

$$\begin{aligned}
5^{2^7 x_0} &\equiv 7^{2^7} \equiv -1 \pmod{257} \Rightarrow x_0 = 1, & 7/5^{2^0} &\equiv -50 \pmod{257}; \\
5^{2^6 x_1} &\equiv (-50)^{2^6} \equiv -1 \pmod{257} \Rightarrow x_1 = 1, & -50/5^{2^1} &\equiv -2 \pmod{257}; \\
5^{2^5 x_2} &\equiv (-2)^{2^5} \equiv 1 \pmod{257} \Rightarrow x_2 = 0; \\
5^{2^4 x_3} &\equiv (-2)^{2^4} \equiv 1 \pmod{257} \Rightarrow x_3 = 0; \\
5^{2^3 x_4} &\equiv (-2)^{2^3} \equiv -1 \pmod{257} \Rightarrow x_4 = 1, & -2/5^{2^4} &\equiv -16 \pmod{257}; \\
5^{2^2 x_5} &\equiv (-16)^{2^2} \equiv 1 \pmod{257} \Rightarrow x_5 = 0; \\
5^{2x_6} &\equiv (-16)^2 \equiv -1 \pmod{257} \Rightarrow x_6 = 1, & -16/5^{2^6} &\equiv -1 \pmod{257}; \\
5^{x_7} &\equiv -1 \pmod{257} \Rightarrow x_7 = 1, & [-1/5^{2^7} &\equiv 1 \pmod{257}].
\end{aligned}$$

(L'ultimo passaggio, scritto fra parentesi, appare qui per completezza, ma naturalmente non serve eseguirlo.) Concludiamo che il logaritmo discreto di 7 in base 5 in \mathbb{F}_{257} è $2^0 + 2^1 + 2^4 + 2^6 + 2^7 = 211$.

Notate che, nell'esempio, ogni bit x_i è stato determinato dall'aver calcolato $(-1)^{x_i} = \pm 1$, e -1 è una (anzi l'unica) radice quadrata primitiva dell'unità in \mathbb{F}_{257} . Più in generale, se p è un primo che divide $q - 1$, diciamo $p^\alpha \mid q - 1$, $b \in \mathbb{F}_q$ è un elemento di ordine p^α , e c è una sua potenza, diciamo $b^x = c$, un algoritmo analogo ci permetterebbe di calcolare successivamente le cifre dell'esponente x in base p ; il ruolo di ± 1 sarebbe giocato dalle p radici p -esime dell'unità in \mathbb{F}_q , che bisognerebbe calcolarsi preventivamente.

In generale, se $q - 1 = \prod_p p^\alpha$, il teorema cinese dei resti permette di ricostruire $x \pmod{q - 1}$ dai valori $x \pmod{p^\alpha}$ per ciascun divisore primo p di $q - 1$, quindi occupiamoci di come determinare questi ultimi. Per ciascun divisore primo p di $q - 1$ bisogna calcolarsi preventivamente $b^{j(q-1)/p}$ per $j = 0, \dots, p - 1$ (che sono le radici p -esime dell'unità in \mathbb{F}_q). Una volta preparate delle tabelle con queste informazioni si è pronti per calcolare il logaritmo discreto in base b (un fissato generatore di \mathbb{F}_q) di qualsiasi elemento c di \mathbb{F}_q^* , cioè l'intero $0 \leq x < q - 1$ tale che $b^x = c$.

Per un primo fissato p scriviamo il resto di x modulo p^α in base p , cioè $x \equiv x_0 + x_1 p + \dots + x_{\alpha-1} p^{\alpha-1} \pmod{p^\alpha}$ con $0 \leq x_i < p$. Calcoliamo

$$c^{(q-1)/p} = b^{x(q-1)/p} = b^{x_0(q-1)/p},$$

ed usiamo la tabella preparata in precedenza per ricavarne x_0 . Quindi calcoliamo cb^{-x_0} , che è una potenza di b con esponente multiplo di p . Questo è una p -esima potenza, e quindi $(cb^{-x_0})^{(q-1)/p} = 1$. Ora calcoliamo

$$(cb^{-x_0})^{(q-1)/p^2} = b^{(x-x_0)(q-1)/p^2} = b^{x_1(q-1)/p},$$

ed usiamo la tabella preparata in precedenza per ricavarne x_1 . Quindi calcoliamo $cb^{-x_0} \cdot b^{-x_1p} = cb^{-x_0-x_1p}$, che è una p^2 -esima potenza. Ora calcoliamo

$$(cb^{-x_0-x_1p})^{(q-1)/p^3} = b^{(x-x_0-x_1p)(q-1)/p^3} = b^{x_2(q-1)/p},$$

e usando la tabella ne ricaviamo x_2 . E cosí via.

ESEMPIO 3.10. Consideriamo il campo primo \mathbb{F}_q , con $q = 251$. Si verifica che $b = 11$ è un generatore, cioè una radice primitiva modulo 251. Essendo $251 = 2 \cdot 5^3$, per calcolare i logaritmi in base b dobbiamo preliminarmente calcolare $b^{j(q-1)/p}$ in \mathbb{F}_q per $p = 2, 5$ e per $0 \leq j < p$. Per $p = 2$ abbiamo $11^0 \equiv 1 \pmod{251}$ e $11^{250/2} \equiv -1 \pmod{251}$ (e qui non serve far conti, lo sappiamo dalla Proposizione di Eulero 2.11), mentre per $p = 5$ calcoliamo

$$11^0 \equiv 1 \pmod{251},$$

$$11^{250/5} \equiv -32 \pmod{251},$$

$$11^{2 \cdot 250/5} \equiv 20 \pmod{251},$$

$$11^{3 \cdot 250/5} \equiv 113 \pmod{251},$$

$$11^{4 \cdot 250/5} \equiv -102 \pmod{251}.$$

Ora siamo pronti per calcolare qualsiasi logaritmo discreto in base 11 in \mathbb{F}_{251} . Ad esempio, calcoliamo il logaritmo di 172, cioè l'intero $0 \leq x < 250$ tale che $11^x \equiv 172 \pmod{251}$.

Per $p = 2$ abbiamo solo una cifra binaria da scoprire, cioè il resto di x modulo 2 (cioè la parità di x). Calcolando $172^{250/2} \equiv -1 \pmod{251}$ scopriamo che tale cifra è 1, cioè che x è dispari. (Naturalmente in questo caso speciale potevamo anche farlo calcolando il simbolo di Legendre $\left(\frac{251}{172}\right) = -1$.) Per $p = 5$ scriviamo $x \equiv x_0 + 5x_1 + 25x_2 \pmod{125}$. Eseguiamo i seguenti calcoli, utilizzando la tabella precompilata per ricavare ciascun x_i :

$$172^{250/5} \equiv 20 \pmod{251} \Rightarrow x_0 = 2, \quad 172/11^{2 \cdot 5^0} \equiv -94 \pmod{251};$$

$$(-94)^{250/5^2} \equiv -102 \pmod{251} \Rightarrow x_1 = 4, \quad -94/11^{4 \cdot 5^1} \equiv 102 \pmod{251};$$

$$102^{250/5^3} \equiv 113 \pmod{251} \Rightarrow x_2 = 3, \quad [102/11^{3 \cdot 5^2} \equiv 1 \pmod{251}].$$

(Come nell'esempio precedente, l'ultimo passaggio, scritto fra parentesi, è chiaramente inutile.) Concludiamo che $x \equiv 1 \pmod{2}$ e $x \equiv 2 + 4 \cdot 5 + 3 \cdot 5^2 = 97 \pmod{5^3}$, perciò $x \equiv 97 \pmod{251}$.

CAPITOLO 4

Test di primalità

4.1. Premesse

In vari metodi di crittografia analizzati c'è la necessità di scegliere dei numeri primi. Un modo semplice per procurarsi un numero primo è il seguente. Ci sono programmi che danno numeri interi random in un certo range. Ne scegliamo uno, diciamolo n . Se n è pari lo aumento di 1 e controllo se è primo, aumento così di 2 in 2 fino a quando non trovo un primo: mi aspetterò di dover eseguire, mediamente, $O(\log n)$ passi. (Notate che se due primi consecutivi nei dintorni di n distassero esattamente $\log n$, avrei la certezza di trovarne uno dopo al massimo $\frac{1}{2} \log n$ passi (visto che sto testando solo i numeri dispari), ma sappiamo che $\log n$ è solo la distanza media di due primi consecutivi in quella zona.) Quindi mi serve un *test di primalità*. Quello che noi vedremo è un *test probabilistico*, che risponde con certezza quando un numero non è primo, mentre mostra che è primo solo con una certa probabilità. (Per questo motivo dovrebbe piuttosto essere chiamato un *test di compositezza*.) Se un numero supera alcuni di questi test e si vede che c'è una bassa probabilità di errore, per molti scopi ci si può accontentare. Una volta passati questi test ci sarebbero dei veri e propri certificati di primalità che garantiscono in un tempo accettabile che un intero è primo (anche se sono molto più lenti di un test probabilistico).

Un modo ovvio per vedere se n è primo è provare a dividere per $2, 3, \dots, \lfloor \sqrt{n} \rfloor$. Naturalmente ciò permette anche di fattorizzare completamente n nel caso esso non sia primo, ma abbiamo visto nell'Esempio 1.15 che per farlo serve un tempo altissimo, $O(n^{(1/2)+\epsilon})$ operazioni bit.

4.2. Pseudoprimi

DEFINIZIONE 4.1. Un intero dispari n si dice *pseudoprimo* rispetto alla base b , se vale

$$b^{n-1} \equiv 1 \pmod{n}.$$

Chiaramente ciò può accadere solo se $(b, n) = 1$. Vediamo da dove nasce questa definizione. Ricordiamo il

TEOREMA (Piccolo Teorema di Fermat). *Sia n un primo e b un intero. Se $(b, n) = 1$, allora*

$$b^{n-1} \equiv 1 \pmod{n}$$

Uno pseudoprimo è un intero che pretenderebbe di essere primo passando solo il test $b^{n-1} \equiv 1 \pmod{n}$ per un certo b .

Quindi se vogliamo verificare se n è primo, se $b^{n-1} \not\equiv 1 \pmod{n}$ per qualche b con $(b, n) = 1$, allora n non è primo. Se $b^{n-1} \equiv 1 \pmod{n}$, allora n è pseudoprimo rispetto alla base b .

Notiamo che se n è primo, allora per il Piccolo Teorema di Fermat n è pseudoprimo per ogni base b . Inoltre ogni n dispari è pseudoprimo rispetto alle basi 1 e -1 .

DEFINIZIONE 4.2. Sia n un intero dispari composito. Allora n si dice *numero di Carmichael* se

$$b^{n-1} \equiv 1 \pmod{n} \quad \text{per ogni } b \text{ tale che } (b, n) = 1.$$

Cioè un numero di Carmichael è un numero non primo, ma pseudoprimo rispetto a qualunque base.

ESEMPIO 4.3. Verifichiamo che 91 è pseudoprimo rispetto alla base 3, ma non è pseudoprimo rispetto alla base 2 (infatti $91 = 7 \cdot 13$ non è primo).

Cominciamo con il verificare che $3^{90} \equiv 1 \pmod{91}$. Possiamo procedere in due modi. Poiché $90 = 2^6 + 2^4 + 2^3 + 2$ otteniamo:

$$3^{90} = 3^{2^6} \cdot 3^{2^4} \cdot 3^{2^3} \cdot 3^2 \equiv 81 \cdot 81 \cdot 9 \cdot 9 \equiv 1 \pmod{91}$$

(Abbiamo già eseguito questo calcolo in altro modo nella Sottosezione 1.3.4.) Da un altro punto di vista possiamo sfruttare il fatto di conoscere la scomposizione di 91

$$\begin{aligned} U(\mathbb{Z}/91\mathbb{Z}) &\cong U(\mathbb{Z}/7\mathbb{Z}) \times U(\mathbb{Z}/13\mathbb{Z}) \\ &\cong C_6 \times C_{12} \end{aligned}$$

dove l'isomorfismo tra anelli è trasformato in isomorfismo tra gruppi moltiplicativi. Facciamo i conti sfruttando l'isomorfismo, per cui $3^{90} = (3^{90}, 3^{90})$. Riducendo i due esponenti modulo 6 e modulo 12 rispettivamente otteniamo

$$3^{90} = (3^0, 3^6) = (1, 3^6)$$

Quindi otteniamo che

$$\begin{cases} 3^{90} \equiv 1 \pmod{7} \\ 3^{90} \equiv 3^6 \pmod{13} \end{cases} \quad \text{ovvero} \quad \begin{cases} 3^{90} \equiv 1 \pmod{7} \\ 3^{90} \equiv 1 \pmod{13} \end{cases}$$

Di conseguenza $3^{90} \equiv 1 \pmod{91}$.

Verifichiamo ora che $2^{90} \not\equiv 1 \pmod{91}$. Procediamo nel secondo dei due modi visti, sfruttando l'isomorfismo. Quindi $2^{90} = (1, 2^6)$ e dobbiamo risolvere il sistema

$$\begin{cases} 2^{90} \equiv 1 \pmod{7} \\ 2^{90} \equiv 2^6 \pmod{13} \end{cases} \quad \text{ovvero} \quad \begin{cases} 2^{90} \equiv 1 \pmod{7} \\ 2^{90} \equiv 64 \pmod{13} \end{cases}$$

Quindi $2^{90} \equiv 64 \pmod{91}$.

PROPOSIZIONE 4.4. Sia n un intero composito (cioè $n \neq 1$ e non primo). Se n non è pseudoprimo per almeno una base b , allora n non è pseudoprimo rispetto ad almeno la metà delle basi possibili (che sono le $\varphi(n)$ basi b con $0 < b < n$ e $(b, n) = 1$).

DIMOSTRAZIONE. Sia $\mathcal{B} = \{b \in (\mathbb{Z}/n\mathbb{Z})^* : b^{n-1} = 1\}$ l'insieme delle basi per cui n è pseudoprimo (viste modulo n). Osserviamo che \mathcal{B} è un sottogruppo proprio di $(\mathbb{Z}/n\mathbb{Z})^*$. Infatti è un sottogruppo in quanto, dati $a, b \in \mathcal{B}$, cioè $a^{n-1} = b^{n-1} = 1$, segue che $(ab)^{n-1} = 1$, quindi anche $ab \in \mathcal{B}$, e $(a^{-1})^{n-1} = (a^{n-1})^{-1} = 1$, quindi $a^{-1} \in \mathcal{B}$. Inoltre per ipotesi esiste almeno una base per cui n non è pseudoprimo e che quindi non è in \mathcal{B} . Di conseguenza

$$|\mathcal{B}| = \frac{\varphi(n)}{|U(\mathbb{Z}/n\mathbb{Z}) : \mathcal{B}|} \leq \frac{\varphi(n)}{2}$$

□

Diamo ora una caratterizzazione dei numeri di Carmichael.

PROPOSIZIONE 4.5. *Sia n un numero dispari composito. Allora n è un numero di Carmichael se e solo se n è libero da quadrati e $p-1$ divide $n-1$ per ogni divisore primo p di n .*

DIMOSTRAZIONE. Vediamo le due implicazioni in un colpo solo. Sia $n = p_1^{\alpha_1} \dots p_k^{\alpha_k}$ con p_i primi distinti e $p_1 < \dots < p_k$. Ricordiamo che per definizione n è un numero di Carmichael sse n non è primo e $b^{n-1} \equiv 1 \pmod{n} \forall b \in U(\mathbb{Z}/n\mathbb{Z})$, ovvero se l'esponente del gruppo $U(\mathbb{Z}/n\mathbb{Z})$ divide $n-1$. Abbiamo visto in precedenza che l'esponente del gruppo $U(\mathbb{Z}/n\mathbb{Z})$ è il minimo comune multiplo di

$$\varphi(p_1^{\alpha_1}) = (p_1 - 1)p_1^{\alpha_1 - 1}, \dots, \varphi(p_k^{\alpha_k}) = (p_k - 1)p_k^{\alpha_k - 1}.$$

Quindi n è di Carmichael sse $n-1$ è multiplo di ogni $(p_i - 1)p_i^{\alpha_i - 1}$, ovvero se $(p_i - 1)p_i^{\alpha_i - 1}$ divide $(n-1)$ per ogni $i = 1 \dots k$. Poiché $p_i \mid n$ segue che p_i non divide $n-1$, quindi n è di Carmichael sse

$$\begin{cases} \alpha_i = 1 & \forall i = 1 \dots k \\ p_i - 1 \mid n - 1 & \forall i = 1 \dots k \end{cases}$$

□

PROPOSIZIONE 4.6. *Un numero di Carmichael è il prodotto di almeno tre primi distinti.*

DIMOSTRAZIONE. Supponiamo per assurdo che un numero di Carmichael n sia prodotto di due primi: $n = pq$ con $p < q$ distinti. Dalla proposizione precedente sappiamo che $p-1$ e $q-1$ dividono $n-1$. Ora $n-1 = (p-1)(q-1) + (p-1) + (q-1)$, quindi $p-1 \mid n-1$ implica $p-1 \mid q-1$ e $q-1 \mid n-1$ implica $q-1 \mid p-1$. Di conseguenza $p-1 = q-1$ e $p = q$ in contraddizione con le nostre ipotesi. □

ESEMPIO 4.7. Il numero $561 = 3 \cdot 11 \cdot 17$ è un numero di Carmichael. Ovviamente per verificarlo non usiamo la definizione, ma la caratterizzazione data dalla Proposizione 4.5. Innanzitutto n è libero da quadrati in quanto prodotto di primi distinti, inoltre $p-1 \mid n-1$ per ogni divisore primo p di n , infatti

$$3 - 1 = 2 \mid 561 - 1 = 560$$

$$11 - 1 = 10 \mid 560$$

$$17 - 1 = 16 \mid 560$$

ESEMPIO 4.8. Essendo

$$\begin{aligned} pqr - 1 &= (p-1)(q-1)(r-1) \\ &+ (p-1)(q-1) + (p-1)(r-1) + (q-1)(r-1) \\ &+ (p-1) + (q-1) + (r-1), \end{aligned}$$

un intero $n = pqr$ prodotto di tre primi distinti è un numero di Carmichael se e solo se

$$p \mid qr - 1, \quad q \mid pr - 1, \quad r \mid pq - 1.$$

4.3. Pseudoprimi di Eulero

Il test di primalità degli pseudoprimi ha come difetto l'esistenza dei numeri di Carmichael. Vediamo ora un test più raffinato, che non soffre di questo difetto.

DEFINIZIONE 4.9. Un intero dispari n si dice *pseudoprimo di Eulero* rispetto alla base b con $(b, n) = 1$ se

$$b^{(n-1)/2} \equiv \left(\frac{b}{n}\right) \pmod{n}$$

Notiamo innanzitutto che se n è primo allora n è pseudoprimo di Eulero rispetto ad ogni base b . Inoltre se n è pseudoprimo di Eulero rispetto alla base b , allora n è pseudoprimo rispetto alla stessa base, ma non vale il viceversa. Infatti se n è pseudoprimo di Eulero otteniamo

$$\begin{aligned} b^{(n-1)/2} &\equiv \left(\frac{b}{n}\right) \pmod{n} \\ (b^{(n-1)/2})^2 &\equiv (\pm 1)^2 \pmod{n} & b^{n-1} &\equiv 1 \pmod{n} \end{aligned}$$

e quindi n è pseudoprimo.

Per dimostrare che non vale l'implicazione inversa diamo un controesempio.

ESEMPIO 4.10. $n = 341$ è pseudoprimo rispetto alla base 2 (si può vedere che è il più piccolo pseudoprimo rispetto alla base 2), ma non è pseudoprimo di Eulero rispetto alla base 2. Infatti si verifica che $2^{340} \equiv 1 \pmod{341}$ utilizziamo la scomposizione $341 = 11 \cdot 31$ (si potrebbe anche fare il conto direttamente). Quindi

$$U(\mathbb{Z}/341\mathbb{Z}) \cong U(\mathbb{Z}/11\mathbb{Z}) \times U(\mathbb{Z}/31\mathbb{Z})$$

Risolvendo il sistema

$$\begin{cases} 2^{340} \equiv 2^0 \equiv 1 \pmod{11} \\ 2^{340} \equiv 2^{10} \equiv 1 \pmod{31} \end{cases}$$

otteniamo che $2^{340} \equiv 1 \pmod{341}$. D'altra parte $\left(\frac{2}{341}\right) = -1$, mentre per calcolare $2^{\frac{340}{2}} = 2^{170}$ sfruttiamo l'isomorfismo precedente.

$$\begin{cases} 2^{170} \equiv 2^0 \equiv 1 \pmod{11} \\ 2^{170} \equiv 2^{20} = (2^{10})^2 \equiv 1 \pmod{31} \end{cases}$$

Quindi $2^{170} \equiv 1 \pmod{341} \not\equiv \left(\frac{2}{341}\right) = -1$

Il *Test di primalità di Solovay-Strassen* si basa sull'essere pseudoprimo di Eulero.

Test di primalità di Solovay-Strassen. Sia n un intero dispari di cui vogliamo stabilire se è primo. Scelte delle basi b in maniera random si controlla se vale

$$b^{(n-1)/2} \equiv \left(\frac{b}{n}\right) \pmod{n}$$

Se troviamo un b per cui tale congruenza non è verificata, allora n non è pseudoprimo di Eulero rispetto a b e quindi non è primo.

Notiamo che $b^{(n-1)/2}$ può essere calcolato efficientemente in $O(\log^3 n)$ operazioni bit, e che $\left(\frac{b}{n}\right)$ può essere calcolato con la Legge di Reciprocità Quadratica in $O(\log^2 n)$ operazioni bit.

PROPOSIZIONE 4.11. *Assumendo che n non sia primo allora*

$$b^{(n-1)/2} \not\equiv \left(\frac{b}{n}\right) \pmod{n}$$

per almeno metà delle basi b possibili.

CENNO DI DIMOSTRAZIONE. Consideriamo l'insieme

$$\mathcal{B} = \left\{ b \in U(\mathbb{Z}/n\mathbb{Z}) : b^{(n-1)/2} \equiv \left(\frac{b}{n}\right) \pmod{n} \right\}$$

e dimostriamo che si tratta di un sottogruppo di $U(\mathbb{Z}/n\mathbb{Z})$. Possiamo fare i calcoli oppure ragionare in questo modo. Riprendiamo il caso degli pseudoprimi. Avevamo definito l'insieme

$$\mathcal{A} = \{ b \in U(\mathbb{Z}/n\mathbb{Z}) : b^{n-1} = 1 \}$$

Consideriamo l'endomorfismo

$$\begin{aligned} \psi : U(\mathbb{Z}/n\mathbb{Z}) &\rightarrow U(\mathbb{Z}/n\mathbb{Z}) \\ b &\mapsto b^{n-1} \end{aligned}$$

il cui nucleo è $\ker(\psi) = \mathcal{A}$ che è quindi un sottogruppo.

Tornando al nostro caso consideriamo le mappe

$$\begin{aligned} \psi_1 : U(\mathbb{Z}/n\mathbb{Z}) &\rightarrow \{\pm 1\} \\ b &\mapsto \pm 1 \\ \psi_2 : U(\mathbb{Z}/n\mathbb{Z}) &\rightarrow U(\mathbb{Z}/n\mathbb{Z}) \\ b &\mapsto b^{(n-1)/2} \end{aligned}$$

\mathcal{B} è l'insieme degli elementi di $U(\mathbb{Z}/n\mathbb{Z})$ su cui le due mappe coincidono. Ora ψ_1 è un omomorfismo e ψ_2 è un endomorfismo. L'insieme su cui due omomorfismo coincidono non è in generale un sottogruppo, ma lo è nel caso abeliano (come il nostro).

In maniera differente potevamo considerare la mappa $b \mapsto b^{(n-1)/2} \left(\frac{b}{n}\right)$. Poiché la condizione $b^{(n-1)/2} = \left(\frac{b}{n}\right)$ è equivalente a $b^{(n-1)/2} \left(\frac{b}{n}\right) = 1$, il nucleo di tale mappa è \mathcal{B} . Quindi \mathcal{B} è un sottogruppo di $U(\mathbb{Z}/n\mathbb{Z})$.

Dimostriamo ora che \mathcal{B} è un sottogruppo proprio, cioè che se n non è primo esistono effettivamente dei b tali che

$$(2) \quad b^{(n-1)/2} \not\equiv \left(\frac{b}{n}\right) \pmod{n}.$$

Per cominciare, se esiste un numero primo p tale che $p^2 \mid n$, allora basta prendere $b = 1 + n/p$ in (2). Infatti se $n = p_1 p_2 p_3 \dots$ è la scrittura di n come prodotto di primi, con $p_1 = p_2 = p$, allora

$$\left(\frac{b}{n}\right) = \prod_i \left(\frac{b}{p_i}\right) = \prod_i \left(\frac{1}{p_i}\right) = 1,$$

dato che $b \equiv 1 \pmod{p_i}$ per ogni i . D'altra parte

$$b^j = \left(1 + \frac{n}{p}\right)^j \equiv 1 + j \frac{n}{p} \pmod{n}$$

è congruo a 1 solo quando $p \mid j$. Ma p non divide $(n-1)/2$. Dunque il termine di sinistra di (2) non è 1, mentre quello di destra lo è 1.

Supponiamo dunque che $n = p_1 p_2 p_3 \dots$ sia prodotto di primi distinti, e sia $p = p_1$. Supponiamo di trovare b tale che

$$(3) \quad \begin{cases} \left(\frac{b}{p}\right) = -1 \\ b \equiv 1 \pmod{n/p}. \end{cases}$$

Allora b soddisfa (2). Infatti

$$\left(\frac{b}{n}\right) = \left(\frac{b}{p}\right) \cdot \prod_{i>1} \left(\frac{b}{p_i}\right) = (-1) \cdot \prod_{i>1} \left(\frac{1}{p_i}\right) = -1,$$

dato che $b \equiv 1 \pmod{p_i}$ per $i > 1$. D'altra parte

$$b^{(n-1)/2} \equiv 1 \pmod{n/p},$$

dunque non può certo essere

$$b^{(n-1)/2} \equiv -1 \pmod{n}.$$

Per trovare b che soddisfi (3) basta risolvere

$$\begin{cases} b \equiv a \pmod{p} \\ b \equiv 1 \pmod{n/p}, \end{cases}$$

ove a non è un resto quadratico modulo p . Notate che $(p, n/p) = 1$, dunque il sistema di congruenze ha soluzione. \square

OSSERVAZIONI. (1) Se troviamo una base b per cui $b^{n-1} \equiv 1 \pmod{n}$ (cioè rispetto alla quale n è pseudoprimo) ma $b^{(n-1)/2} \not\equiv \pm 1 \pmod{n}$, possiamo concludere che n non è primo, senza bisogno di calcolare $\left(\frac{b}{n}\right)$.

- (2) Se si verifica l'evento appena descritto, possiamo immediatamente fattorizzare n , in quanto abbiamo scoperto una radice di 1 modulo p diversa da ± 1 modulo p (e precisamente $b^{(n-1)/2}$). Questo evento in generale sarà molto raro (vale a dire, applicando il test degli pseudoprimi ad un intero composito grande n , quasi sicuramente scopriremo che n non è primo al primo passo), e quindi questo non è un metodo per fattorizzare un intero n .

Tuttavia nel metodo RSA vorremo fare in modo che tale evento sia il più raro possibile: un modo per assicurarsene è minimizzare il numero di basi rispetto a cui n è pseudoprimo.

Usando il Teorema cinese dei resti è facile mostrare (come abbiamo fatto a lezione) che il numero di basi b (sempre intese prime con $n = pq$) rispetto a cui n è pseudoprimo è $(p-1, q-1)^2$. Questo è il motivo per cui nel metodo RSA i primi p e q andranno scelti in modo che $(p-1, q-1)$ sia piccolo.

4.4. Pseudoprimi forti

Sia n pseudoprimo, per cui $b^{n-1} \equiv 1 \pmod{n}$. Estraiamo radici quadrate successive di questa congruenza, cioè eleviamo alla $(n-1)/2, (n-1)/4, \dots$ finché riusciamo. Se n è primo otteniamo solo ± 1 (dato che esistono solo due radici quadrate di 1 modulo un primo).

ESEMPIO 4.12. Consideriamo $n = 341$. Da un esempio precedente sappiamo già che $2^{340} \equiv 1 \pmod{341}$. Calcoliamo ora $2^{340/2} = 2^{170} \pmod{341}$. Se ottenessimo un risultato diverso da ± 1 avremmo che 1 ha almeno 4 radici quadrate differenti e quindi 341 non potrebbe essere un primo. In realtà abbiamo $2^{170} \equiv 1 \pmod{341}$. Invece di confrontarlo con il simbolo di Jacobi $\left(\frac{2}{341}\right)$ come nel test di Solovay-Strassen, visto che l'esponente ancora pari, possiamo dimezzarlo di nuovo e calcolare $2^{341/4} = 2^{85} \equiv 32 \pmod{341}$. Essendo questo $\not\equiv \pm 1$, concludiamo che 341 non è primo. (Anzi, avendo trovato che $341 \mid 32^2 - 1 = (32-1)(32+1)$, in questo caso riusciamo addirittura a fattorizzare $341 = 11 \cdot 31$.)

Su tale concetto si basa il *Test di Miller-Rabin*.

Test di primalità di Miller-Rabin. Sia n un intero e b tale che $(b, n) = 1$. Abbiamo due possibilità

- Se $b^{n-1} \not\equiv 1 \pmod{n}$ possiamo concludere che n non è un primo.
- Se $b^{n-1} \equiv 1 \pmod{n}$, controlliamo $b^{(n-1)/2}$ e abbiamo le seguenti possibilità. Se

$$b^{(n-1)/2} \not\equiv \pm 1 \pmod{n}$$

possiamo concludere che n non è un primo. Se

$$b^{(n-1)/2} \equiv -1 \pmod{n}$$

il test non ci dà informazioni, per cui dobbiamo considerare un differente valore di b . Se

$$b^{(n-1)/2} \equiv 1 \pmod{n}$$

dobbiamo ancora distinguere due casi. Se $(n-1)/4 \notin \mathbb{Z}$, come nel caso precedente il test non ci da informazioni e dobbiamo cambiare b , se invece $(n-1)/4 \in \mathbb{Z}$ controlliamo il valore di $b^{(n-1)/4}$ ripetendo i passi fatti per $b^{(n-1)/2}$.

DEFINIZIONE 4.13. Sia n un intero dispari composto. Scriviamo $n-1 = 2^s t$ con t dispari. Sia $b \in U(\mathbb{Z}/n\mathbb{Z})$. Allora n si dice *pseudoprimo forte* rispetto alla base b se vale

$$\begin{aligned} b^{(n-1)/2^s} &\equiv 1 \pmod{n}, & \text{oppure} \\ b^{(n-1)/2^r} &\equiv -1 \pmod{n} & \text{per qualche } 0 < r \leq s. \end{aligned}$$

Dunque, per definizione, le basi b rispetto alle quali n è uno pseudoprimo forte sono esattamente quelle che non danno informazioni nel test di Miller-Rabin. Inoltre se n è uno pseudoprimo forte rispetto alla base b , allora è pseudoprimo rispetto alla base b , mentre non è detto che sia uno pseudoprimo di Eulero rispetto a b .

Si può dimostrare un fatto analogo alle Proposizioni 4.4 e 4.11 (la dimostrazione è però piú difficile, si veda [Kob94]): un intero composto n è uno pseudoprimo forte rispetto ad piú un quarto delle possibili basi b . In particolare, cosí come per gli pseudoprimi di Eulero, nemmeno per gli pseudoprimi forti non esistono eccezioni analoghe ai numeri di Carmichael per gli pseudoprimi. Inoltre il test di Miller-Rabin è piú efficiente di quello di Solovay-Strassen: a parte non dover calcolare simboli di Jacobi, per ogni base b provata nel test di Miller-Rabin ho una probabilità al massimo del 25% di non ottenere informazioni. In altre parole, se n non è primo, c'è una probabilità almeno del 75% di scoprirlo scegliendo una base random b . Si noti infine che un passaggio del test di Miller-Rabin non comporta davvero alcun calcolo in piú rispetto a quello degli pseudoprimi normali, in quanto le potenze $b^{(n-1)/2^r} \pmod{n}$ vengono comunque ottenute come passi intermedi nel calcolo di $b^{n-1} \pmod{n}$, se si utilizza il metodo dei quadrati ripetuti.

Metodi di fattorizzazione

DEFINIZIONE 5.1. Sia B un intero positivo. Un intero positivo n si dice *B-smooth* se tutti i divisori primi di n sono minori o uguali a B ; n si dice *B-powersmooth* se tutti le potenze di primi che dividono n sono minori o uguali a B .

5.1. Il metodo $p - 1$ di Pollard

Il metodo $p - 1$ di Pollard (da non confondere con il metodo ρ di Pollard) è un metodo di fattorizzazione efficiente per scoprire un fattore primo p di n tale che $p - 1$ sia prodotto di primi “abbastanza piccoli”, nel senso preciso che $p - 1$ sia *B-powersmooth* per un B non troppo grande. È quindi un metodo di fattorizzazione *specializzato* piuttosto che *generico*, cioè funziona bene su interi n con proprietà particolari (un po’ come il metodo di Fermat che funziona bene solo se n ha un fattore vicino a \sqrt{n}), ed in effetti si può mostrare che nel caso generico esso può impiegare fino a $O(\sqrt{n})$ operazioni bit, cioè sostanzialmente lo stesso costo delle divisioni per tentativi.

Però nei casi speciali in cui funziona esso è particolarmente efficiente. In particolare, questo giustifica perché nel descrivere il metodo RSA abbiamo raccomandato di scegliere p e q in modo che ciascuno di $p - 1$ e $q - 1$ *abbia almeno un fattore primo grande*. (Questa vaga specifica andrebbe comunque precisata, per tener conto di raffinamenti del metodo di Pollard come qui descritto.)

Supponiamo che n non sia primo, e sia p un fattore primo di n . Allora per ogni intero a primo con p , e per ogni intero k multiplo di $p - 1$ avremo $a^k \equiv 1 \pmod{p}$, e quindi $(a^k - 1, n) > 1$. Se riusciamo a trovare un multiplo k non troppo grande di $p - 1$, questo ci permette di trovare un fattore proprio di n , ad eccezione del caso in cui valga anche $a^k \equiv 1 \pmod{n}$. Rimane il problema di trovare un k che va bene, ovviamente senza conoscere p . (Naturalmente $k = \lceil \sqrt{n} \rceil!$ è multiplo di p , ma è troppo grande!)

Se $p - 1$ è *B-powersmooth*, per un B relativamente piccolo allora $p - 1$ divide il minimo comune multiplo k_B di tutti i numeri minori o uguali a B , e quindi quello è un possibile valore di k . In pratica si calcola tale minimo comune multiplo k_B ricorsivamente, partendo da B piccolo e poi facendolo aumentare di uno ad ogni passo, come spiegato nel seguente esempio. A ciascun passo, il minimo comune multiplo k corrispondente a B si ottiene da quello per $B - 1$ moltiplicandolo per r se B è una potenza di un primo r (e lasciandolo invariato altrimenti). Allo stesso tempo, per una base a fissata inizialmente (ad esempio $a = 2$ o 3) si calcola $a^{k_B} \pmod{n}$ come $(a^{k_{B-1}})^r \pmod{n}$, se B è una potenza di un primo r (mentre $a^{k_B} = a^{k_{B-1}}$ altrimenti). Dunque il singolo passo si fa in $O(\log^3(B))$ operazioni

bit. Rimane poi da calcolare il massimo comun divisore $(a^k - 1, n)$, ma non è necessario farlo ad ogni passo, basta farlo ogni tanto, ad esempio ogni 20 passi, o ogni 100 (riservandosi eventualmente di tornare indietro e farlo piú spesso, avendo salvato dei valori intermedi di $a^k \pmod{n}$, si veda il seguente esempio).

ESEMPIO 5.2. Applichiamo il metodo $p - 1$ di Pollard al numero $n = 265651$, con $B = 10$ (o $B = 9$, che non fa differenza, perché 10 non è una potenza di un primo). Scegliamo $a = 2$ come base di partenza, e la eleviamo, modulo n , al minimo comune multiplo di tutti i numeri minori o uguali a B , facendo crescere B progressivamente. In questo modo, ogni volta che B è una potenza di un primo r basta elevare all'esponente r la potenza ottenuta al passo precedente, mentre non serve fare nulla quando B non è una potenza di un primo. Ad ogni passo calcoliamo il massimo comun divisore di $a^k - 1$ con n per vedere se troviamo un fattore proprio di n . In realtà basta farlo una volta ogni tanto, ma su questo commenteremo alla fine.

$$\begin{array}{llll}
 B = 2 & a^2 = 2^2 \equiv 4 & \pmod{n} & (4 - 1, n) = 1 \\
 B = 3 & (a^2)^3 = 4^3 \equiv 64 & \pmod{n} & (64 - 1, n) = 1 \\
 B = 4 & (a^{2 \cdot 3})^2 = 64^2 \equiv 4096 & \pmod{n} & (4096 - 1, n) = 1 \\
 B = 5 & (a^{2^2 \cdot 3})^5 = 4096^5 \equiv 8790 & \pmod{n} & (8790 - 1, n) = 1 \\
 B = 7 & (a^{2^2 \cdot 3 \cdot 5})^7 = 8790^7 \equiv -56834 & \pmod{n} & (-56834 - 1, n) = 421 \\
 B = 8 & (a^{2^2 \cdot 3 \cdot 5 \cdot 7})^2 = 56834^2 \equiv 53047 & \pmod{n} & (53047 - 1, n) = 421 \\
 B = 9 & (a^{2^3 \cdot 3 \cdot 5 \cdot 7})^3 = 53047^3 \equiv 1 & \pmod{n} & (1 - 1, n) = n
 \end{array}$$

Vediamo che con $B = 7$ scopriamo un fattore proprio di $n = 421 \cdot 631$. Notate che se avessimo calcolato $(a^k - 1, n)$ solo una volta ogni tanto, diciamo ogni volta che B è multiplo di 3, il nostro massimo comun divisore $(a^k - 1, n)$ sarebbe passato di colpo da 1 (quando $B = 6$) a n (quando $B = 9$); in quel raro caso (provocato qui di proposito, scegliendo entrambi i fattori primi $p = 421$ e 631 di n in modo che $p - 1$ divida $k_9 = 2^3 \cdot 3^2 \cdot 5 \cdot 7$) basta tornare indietro e calcolare il massimo comune multiplo anche per qualche altro valore intermedio di B . Raramente può succedere che nessun massimo comun divisore intermedio $(a^k - 1, n)$ sia un fattore proprio di n , ma si vede facilmente che ciò non può accadere per ogni base a (primo con n), quindi in tal caso basta rifare il procedimento con un altro valore di a . Nel caso del numero n qui considerato ciò sarebbe successo, ad esempio, prendendo $a = 5$ anziché 2 come base iniziale; infatti 5 ha ordine $210 = 2 \cdot 5 \cdot 3 \cdot 7$ modulo 431 e $35 = 5 \cdot 7$ modulo 631, e quindi $a^k - 1$ sarebbe divenuto multiplo di 431 e di 631 simultaneamente, al passo $B = 7$.

In questo esempio i due fattori primi di n sono stati scelti relativamente vicini per illustrare un fenomeno particolare, ma il metodo avrebbe fattorizzato, ad esempio, $n = 77901367 = 631 \cdot 123457$, quasi altrettanto efficientemente: sarebbe accaduto al passo $B = 9$. Notate anche che il primo p individuato dal metodo non è necessariamente il fattore piú piccolo di n : scopriremmo al passo $B = 9$ anche il fattore primo 631 di $n = 307 \cdot 631 = 193717$, o di $n = 457 \cdot 631 \cdot 541 \cdot 3001$, ecc.

Segue da quanto visto che tale avrà successo in un tempo $O(q \log^3(n))$, dove q è la più grande potenza di primo che divide $p - 1$. Quindi nel caso peggiore potrà essere $O(n^{\frac{1}{2}+\varepsilon})$, non meglio delle divisioni per tentativi. Ma il metodo è in grado di trovare rapidamente anche fattori primi p molto grandi di n , purché $p - 1$ sia prodotto di varie potenze di primi, che siano relativamente piccole. Esiste un'importante modifica del metodo che ammette la presenza in $p - 1$ di un singolo fattore primo maggiore di B , anche di molto, ma comunque minore di B^2 (si veda [Coh93, 8.8.2] o [Rie94, Cap. 6]). Esistono inoltre varianti del metodo di Pollard che rimpiazzano la richiesta su $p - 1$, che esso sia B -powersmooth, con un'analogia richiesta su $p + 1$, o su $p^2 - p + 1$ o $p^2 + p + 1$, dove il ruolo del gruppo \mathbb{F}_p^* è essenzialmente giocato dal sottogruppo di $\mathbb{F}_{p^2}^*$ o $\mathbb{F}_{p^3}^*$ dell'ordine opportuno. Nei primi anni '80 il metodo $p - 1$ di Pollard e le sue varianti hanno permesso di scoprire fattori p con 15 - 20 cifre decimali di certi numeri n fino a un centinaio di cifre decimali, si veda [Rie94, Cap. 6]. Il metodo di Pollard è anche importante perché su idee simili è basato il metodo di fattorizzazione di Lenstra con le curve ellittiche; quest'ultimo è molto più flessibile delle varie varianti del metodo di Pollard, perché il gruppo utilizzato è il gruppo dei punti di una curva ellittica su \mathbb{F}_p : il suo ordine, a seconda dell'equazione, può assumere molti valori compresi fra $p \pm 2\sqrt{p}$, piuttosto che i pochi valori $p \pm 1$, ecc., del metodo di Pollard e le sue varianti (si veda [Kob94, VI.3]).

5.2. Fattorizzazione di Fermat

Il metodo di fattorizzazione di Fermat si basa sul cercare interi a e b tali che $a^2 \equiv b^2 \pmod{n}$ ma $a \not\equiv \pm b \pmod{n}$. Infatti in tal caso n divide $a^2 - b^2 = (a - b)(a + b)$, ma non entrambi. Un modo di procurarsi a e b tali che $a^2 \equiv b^2 \pmod{n}$ è scegliere $a \geq \sqrt{n}$ e controllare se il resto della divisione di $a^2 \pmod{n}$ è un quadrato. Abbiamo visto in un esercizio che questo funziona bene se n (dispari) è prodotto di due fattori fra loro vicini: si prendono come valori di a interi consecutivi crescenti a partire da $\lceil \sqrt{n} \rceil$, e si fattorizza n non appena il resto di $a^2 \pmod{n}$ è un quadrato b^2 . Infatti se $n = pq$ con $p < q$ vicini (ma non necessariamente primi) abbiamo successo non appena a raggiunge il valore $(p + q)/2$, e quindi dopo un numero di passaggi vicino a $(p + n/p)/2 - \sqrt{n} = (\sqrt{n} - p)^2/2p$, che è piccolo solo se p e q sono molto vicini. Ma se applicato al caso generale, un tale metodo può richiedere un tempo inaccettabile; ad esempio, se $q \approx p^2$ il numero di passi necessari è $\approx \frac{1}{2}n^{2/3}$.

Un drastico miglioramento si ottiene se ci si limita a cercare alcuni resti $a^2 \pmod{n}$, in modo che *il loro prodotto* sia un quadrato. L'idea è di cercarli in modo che siano prodotti di potenze di alcuni primi p_1, \dots, p_h prefissati, ed eventualmente -1 , cioè che siano B -numeri, rispetto alla *base di fattori* $B = \{-1, p_1, \dots, p_h\}$. Le nostre speranze di trovarne naturalmente aumentano se i primi p_i sono piccoli, e così i resti $a^2 \pmod{n}$. Un modo non molto sofisticato per trovare questi ultimi, esemplificato in [Kob94, V.3, Example 9], è provare con $a = \lfloor \sqrt{kn} \rfloor$ e $\lceil \sqrt{kn} \rceil$ con k piccolo, come in [Kob94, V.3, Example 9]. (Questo perché vogliamo che a^2 sia molto vicino ad un multiplo kn di n . Notate che oltre a valori di a poco più

grandi di \sqrt{n} possiamo sfruttare anche valori poco piú piccoli, in quanto ci vanno bene anche resti negativi, avendo incluso -1 nella base di fattori B .) Un modo molto piú efficiente è cercarli utilizzando l'espansione in frazione continua di \sqrt{n} , che produce buone approssimazioni razionali A/B di \sqrt{n} .

Una volta che abbiamo trovato vari resti $a^2 \pmod{n}$ che siano B -numeri, e li abbiamo fattorizzati completamente, dobbiamo moltiplicarne alcuni fra loro in modo da ottenere un quadrato, cioè dobbiamo fare in modo che la somma dei vettori degli esponenti di $-1, p_1, \dots, p_h$ nelle loro fattorizzazioni abbia tutte le entrate pari. Se ci riusciamo, il prodotto dei corrispondenti valori di a è un intero che elevato al quadrato e poi ridotto modulo n è il quadrato di un numero intero esplicitamente calcolato (come prodotto dei p_i), e quindi molto probabilmente permette di fattorizzare n nel solito modo.

Il modo di selezionare alcuni resti $a^2 \pmod{n}$ tali che il prodotto sia un quadrato di un numero intero, come descritto al paragrafo precedente, è di ridurre modulo due i vettori degli esponenti di $-1, p_1, \dots, p_h$ nelle fattorizzazioni trovate dei resti $a^2 \pmod{n}$ che siano B -numeri, e poi cercare una relazione di dipendenza lineare fra essi, con i metodi dell'algebra lineare (ad esempio riduzione di Gauss, cioè con operazioni elementari sulle righe, o metodi piú sofisticati.) Dato che tali vettori hanno $h+1$ componenti, $h+2$ fra loro saranno sicuramente linearmente dipendenti, e quindi $h+2$ resti $a^2 \pmod{n}$ che siano B -numeri sono sufficienti allo scopo (ma ne potrebbero bastare anche meno).

Per maggiori dettagli, ed esempi, si veda [Kob94, V.3].

5.3. Il crivello quadratico (*quadratic sieve*)

Come nel metodo di Fermat generale descritto sopra, l'idea base è quella di calcolare tanti resti $a^2 \pmod{n}$, e selezionare poi fra questi quelli che sono B -numeri rispetto ad una base di fattori B prefissata. Nella versione base, il metodo per procurarsi i resti è, come nel metodo di Fermat piú elementare, calcolare i resti di $a^2 \pmod{n}$ per $a = \lfloor \sqrt{n} \rfloor + 1, \lfloor \sqrt{n} \rfloor + 1, \lfloor \sqrt{n} \rfloor + 2, \dots, \lfloor \sqrt{n} \rfloor + M$. Qui il numero M di valori di a considerati sarà scelto relativamente piccolo rispetto a n , certamente (molto) minore di $\sqrt{2n}$, perciò le riduzioni modulo n saranno semplicemente sottrazioni $a^2 - n$. In altre parole, i resti cercati saranno i valori del polinomio quadratico $Q(a) = a^2 - n$ sugli a nell'intervallo considerato. (Il nome del metodo deriva dall'usare il polinomio quadratico Q ; una versione piú sofisticata, il *multiple polynomial quadratic sieve*, utilizza vari polinomi quadratici al posto del solo Q .)

In generale, solo una piccolissima frazione dei numeri $a^2 - n$ saranno B -numeri. Un modo di trovarli sarebbe provare a dividere ciascun $a^2 - n$ successivamente per tutti i primi (con ripetizioni, dato che questi potrebbero dividere il resto con molteplicità maggiore di 1) e vedere se alla fine rimane 1. Invece di farlo in questo modo inefficiente, l'idea del metodo è di utilizzare un *crivello* (cioè un *setaccio*), trovando simultaneamente tutti i resti che sono multipli di una certa potenza di p_i , per un certo $p_i \in B$.

Anzitutto, un primo p_i della nostra base di fattori B può sperare di dividere qualche $a^2 - n$, cioè di soddisfare $n \equiv a^2 \pmod{p_i}$, solo se $\left(\frac{n}{p_i}\right) = 1$. Quindi

inseriranno solo tali primi nella base di fattori utilizzata. In pratica, quindi, la base di fattori $B = \{p_1, \dots, p_h\}$ potrebbe contenere tutti i primi p_i minore di un limite prefissato, e tali che n sia un resto quadratico modulo p_i , e perciò conterrà circa la metà dei primi minori di quel limite. (Notate che in questa forma base del metodo non serve includere -1 in B , perché i resti $a^2 - n$ prodotti sono positivi; alternativamente, si potrebbe ammetterli anche negativi estendendo la ricerca a valori di a compresi fra $\sqrt{n} - M$ e $\sqrt{n} + M$, e quindi includere -1 in B .)

Inoltre, un primo p_i della base di fattori B dividerà $a^2 - n$ se e solo se a appartiene ad una delle due progressioni aritmetiche di ragione p_i date da $a \equiv \pm a_{p_i} \pmod{p_i}$, dove $\pm a_{p_i}$ sono le due radici quadrate di n modulo p_i (se $p_i > 2$), che possiamo calcolare agevolmente con il metodo di Tonelli e Shanks (o anche per tentativi per i primi p_i molto piccoli). Un discorso analogo vale rimpiazzando p_i con una sua potenza p_i^α . Dunque calcoliamo preliminarmente, per ogni primo $p_i \in B$ (con p_i dispari), le due radici quadrate $\pm a_{p_i}$ di n modulo p_i . Ora, per ogni a nell'intervallo considerato ed appartenente alle due progressioni aritmetiche $a \equiv \pm a_{p_i} \pmod{p_i}$, possiamo dividere per p_i i corrispondenti resti $a^2 - n$. Poi solleviamo $\pm a_{p_i}$ alle due radici quadrate $\pm a_{p_i^2}$ di n modulo p_i^2 . Le due progressioni aritmetiche $a \equiv \pm a_{p_i^2} \pmod{p_i^2}$ sono contenute nelle precedenti, e per ogni a appartenente ad esse e nell'intervallo considerato possiamo dividere ulteriormente per p_i i corrispondenti resti $a^2 - n$ (che sono già stati divisi una volta per p_i). Continuiamo in questo modo, aumentando progressivamente l'esponente α di p_i^α fintanto che le due progressioni aritmetiche considerate contengono almeno un elemento a nell'intervallo considerato. (Se $2 \in B$, cioè $p_1 = 2$, il che avviene esattamente quando $n \equiv 1 \pmod{8}$, la trattazione per il primo $p_1 = 2$ sarà leggermente diversa in quanto, come sappiamo, n ha quattro radici quadrate modulo p_1^α per $\alpha \geq 3$, anziché due.) Dopo aver ripetuto il procedimento per ogni $p_i \in B$, i resti $a^2 - n$ che sono B -numeri, a cui quindi potremo applicare il processo descritto nella sezione precedente (la riduzione di Gauss sui vettori degli esponenti nelle loro fattorizzazioni), saranno quelli dove $a^2 - n$ ha lasciato come risultato 1 dopo le varie divisioni per i p_i .

Dunque facendo in questo modo si risparmiano molte divisioni inutili rispetto al metodo visto nella sezione precedente, e precisamente si eseguono solo le divisioni di $a^2 - n$ per i vari p_i dettate dal crivello, cioè quelle che danno resto zero. Naturalmente le divisioni che rimangono sono operazioni che si fanno in modo efficiente, ma dato che comunque se ne devono fare moltissime, c'è un ulteriore artificio per diminuire il tempo necessario. Anziché eseguire le divisioni dei resti $a^2 - n$ per i vari p_i , si sottraggono a $\log(a^2 - n)$ successivamente i corrispondenti $\log(p_i)$. Naturalmente, detto in questo modo non sembra certo più efficiente, ma il punto cruciale è che basta eseguire tali sottrazioni usando approssimazioni molto rozze di $\log(a^2 - n)$ e poi verificare se i risultati finali sono vicini a zero (piuttosto che esattamente zero). Ad esempio, un'approssimazione intera di $\log(a^2 - n)$ è più che sufficiente (mentre eventualmente possiamo permetterci di calcolare approssimazioni un poco più precise dei p_i , visto che sono molti meno dei valori $a^2 - n$ da considerare). Il motivo è che, se $a^2 - n$ non è un B -numero, allora il numero ottenuto sottraendo a $\log(a^2 - n)$ i vari $\log p_i$ secondo l'algoritmo è non solo positivo,

ma maggiore o uguale a $\log q$, dove q è il piú piccolo primo che abbiamo lasciato fuori dalla base di fattori B . Essendo $\log q$ molto maggiore di uno, si spiega perché un'approssimazione molto rozza di $\log(a^2 - n)$ è sufficiente. Le sottrazioni che si eseguono in tal modo sono parecchio piú rapide delle corrispondenti divisioni, non tanto perché le sottrazioni sono in generale un po' piú veloci delle divisioni, ma quanto perché vengono eseguite su numeri molto piú piccoli (i logaritmi).

I tempi stimati per fattorizzare n mediante la fattorizzazione di Fermat (con parametri scelti opportunamente, si veda [Kob94, V.3]), il crivello quadratico (cioè il *quadratic sieve*) o mediante il *multiple polynomial quadratic sieve* sono tutti $O(e^{C\sqrt{\log n \log \log n}})$ per certe costanti C (ciascuna migliore della precedente, $C = O(1 + \varepsilon)$ nel terzo caso). L'unico metodo che fa meglio asintoticamente (ma in pratica solo nella parte piú alta o al di là del range dei numeri fattorizzabili al giorno d'oggi) è il *number field sieve*, con un tempo $O(e^{C(\log n)^{1/3}(\log \log n)^{2/3}})$.

Per maggiori dettagli si vedano [Kob94, V.5], [Coh93, 10.4] e [Rie94, p. 204–209].

Bibliografia

- [Coh93] Henri Cohen, *A course in computational algebraic number theory*, Graduate Texts in Mathematics, vol. 138, Springer-Verlag, New York, 1993.
- [CR71] R. Courant and H. Robbins, *Che cos'è la matematica?*, Universale Scientifica Boringhieri, vol. 65–67, Paolo Boringhieri, Torino, 1971.
- [HW79] G. H. Hardy and E. M. Wright, *An introduction to the theory of numbers*, fifth ed., Oxford University Press, Oxford, 1979.
- [IR90] Kenneth Ireland and Michael Rosen, *A classical introduction to modern number theory*, second ed., Graduate Texts in Mathematics, vol. 84, Springer-Verlag, New York, 1990. MR MR1070716 (92e:11001)
- [Kob94] Neal Koblitz, *A course in number theory and cryptography*, second ed., Graduate Texts in Mathematics, vol. 114, Springer-Verlag, New York, 1994. MR 95h:94023
- [Mon85] Peter L. Montgomery, *Modular multiplication without trial division*, Math. Comp. **44** (1985), no. 170, 519–521. MR 86e:11121
- [NZ72] Ivan Niven and Herbert S. Zuckerman, *An introduction to the theory of numbers*, third ed., John Wiley and Sons Inc., New York, 1972.
- [Rie94] Hans Riesel, *Prime numbers and computer methods for factorization*, second ed., Progress in Mathematics, vol. 126, Birkhäuser, Boston, 1994.