Note del corso

di

Algoritmi della Teoria dei Numeri

e Crittografia

Prima parte

Marta Morigi

Nota introduttiva

Queste note sono un riassunto, per non dire un "collage", del materiale riportato in bibliografia e sono state concepite come supporto al corso di Algoritmi della Teoria dei Numeri e Crittografia.

Indice

Capitolo 1

Preliminari

1.1 Stime temporali per l'esecuzione di calcoli

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

$$n = d_{k-1}b^{k-1} + d_{k-2}b^{k-2} + \cdots + d_1 b + d_0$$

dove ciascun d_i è una *cifra* in base b, cioè un simbolo per ciascuno degli interi $0, 1, \ldots, b-1$ (ad esempio $0, 1, \ldots, b-1$ stessi se $b \leq 10$, oppure delle lettere). Se la prima cifra d_{k-1} è diversa da zero, diremo che n è un numero in base b di k cifre e sarà indicato come $(d_{k-1}d_{k-2}\ldots d_1 d_0)_b$.

Esempio: (a) $(11001001)_2 = 201$

(b) Quando $b = 21$ ed usiamo le lettere da A a Z per le cifre da 0 a 20 si ha che $(CASA)_{21} = 590499$.

Osservazione: Si noti 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 $b^{k-1} \leq n < b^k$ (ove $\lfloor\ \rfloor$ denota la funzione "parte intera e log indica il logaritmo naturale $\log_e$).

1.1.2 Conversione da base b a base 10

Dato l'intero $n = (d_{k-1}d_{k-2}\ldots d_1 d_0)_b$, per effettuare il calcolo è conveniente scrivere n nella forma

$$n = (\ldots((d_{k-1}b + d_{k-2})b + d_{k-3})b + \cdots + d_1)b + b_0.$$

In tal modo infatti si effettuano soltanto $k-1$ moltiplicazioni per b e $k-1$ addizioni.

1.1.3 Conversione da base 10 a base b

Se n è un intero, si ottiene la sua ultima cifra in base b, cioè d_0, con il resto della divisione di n per b, poi d_1 come resto della divisione del quoziente precedente per b, e così via (si noti che i resti saranno nulli da un certo punto in poi).

Ad esempio, per convertire $(27686)_{10}$ in base 7 si attua il procedimento seguente:
$27686 = 3955 \cdot 7 + \mathbf{1}$
$3955 = 565 \cdot 7 + \mathbf{0}$
$565 = 80 \cdot 7 + \mathbf{5}$
$80 = 11 \cdot 7 + \mathbf{3}$
$11 = 1 \cdot 7 + \mathbf{4}$
$1 = 0 \cdot 7 + \mathbf{1}$

Quindi $(27686)_{10} = (143501)_7$.

1.1.4 Operazioni bit

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

$$\begin{array}{cccccccc} & {}^{1}1 & {}^{1}0 & {}^{1}1 & {}^{1}1 & 0 & 1 & + \\ & 0 & 1 & 1 & 1 & 0 & 0 & = \\ \hline 1 & 0 & 0 & 1 & 0 & 0 & 1 & \end{array}$$

(le cifre in piccolo indicano i riporti). Possiamo supporre che entrambi i numeri abbiano k cifre, eventualmente aggiungendo degli zeri davanti ad uno di essi, come nell'esempio.

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 richiede k operazioni bit.

Il tempo necessario ad un calcolatore per effettuare un calcolo è essenzialmente proporzionale al numero di operazioni bit. Chiaramente, la costante di proporzionalità, cioè il numero di nanosecondi per operazione bit, dipende dal calcolatore considerato, e non si tiene conto del tempo necessario per operazioni di tipo amministrativo, come accedere alla memoria, ricopiare dati da un posto all'altro, ecc.

Ora vogliamo moltiplicare due interi n ed m nel sistema binario, di k ed l cifre rispettivamente, ad esempio:

$$\begin{array}{cccccccc} & & & 1 & 0 & 0 & 1 & 1 & \times \\ & & & & 1 & 0 & 1 & 1 & = \\ \hline & & & 1 & 0 & 0 & 1 & 1 & \\ & & 1 & 0 & 0 & 1 & 1 & & \\ 1 & 0 & 0 & 1 & 1 & & & & \\ \hline 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & \end{array}$$

Otteniamo $l' \leq l$ righe, una per ogni cifra 1 di m, ciascuna delle quali consiste di una copia di n traslata a sinistra di una certa distanza. Dobbiamo quindi effettuare l^-1 addizioni (si somma la prima riga alla seconda, al risultato si aggunge la terza riga e così via). Si noti che gli addendi hanno in generale un numero di cifre maggiore di k, perchè le somme parziali si allungano, però ciascuna addizione comporta in realtà soltnto k operazioni bit non banali. Per convincersene, basta pensare al nostro esempio. La somma delle prime due righe dà $s = 111001$, e a questo numero dobbiamo sommare 10011000. Per far questo ricopiamo le ultime tre cifre 001 di s, e a quello che resta, cioè 111, aggiungiamo 10011, cioè n, poi alla somma ottenuta 10110 riattacchiamo in coda le tre cifre 001.

In tutto il tempo necessario è di al più $(l' - 1)k < lk$ operazioni bit (non si tiene conto del tempo necessario per traslare le cifre, eliminarne alcune o riattaccarle in coda).

1.1.5 La notazione O-grande

Se $f(n)$ e $g(n)$ sono funzioni (n è un intero non-negativo) a valori reali positivi, diciamo che $f(n) = O(g(n))$ se esistono due costanti B e C tali che $f(n) \leq Cg(n)$ per ogni $n \geq B$. Siccome considereremo funzioni di più variabili (ad esempio il prodotto di due interi), ci serve una definizione leggermente più generale, e precisamente:

Definizione 1.1 *Siano f, g due funzioni a valori reali positivi 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 e C tali che se per ogni $j = 1, \ldots, r$ si ha $n_j > B$, allora*

$$f(n_1, n_2, \ldots, n_r) < Cg(n_1, n_2, \ldots, n_r)$$

In tal caso diremo che f è limitata da *g e scriveremo $f = O(g)$.*

Osservazioni:

1. In questa notazione il segno "=" ha in realtà significato di "< e il simbolo "O" significa "qualche multiplo costante d". Quindi O è una relazione, che gode della proprietà transitiva e riflessiva, ma non simmetrica. Infatti se $f = O(g)$ e $g = O(h)$ segue che $f = O(h)$, ma non che $g = O(f)$.

2. Formalmente in $f = O(g)$ possiamo 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 il fatto che g sarà preferibilmente di facile descrizione.

3. Se $\lim_{n\to\infty} \frac{f(n)}{g(n)}$ esiste finito allora $f = O(g)$.

4. Se $f(n)$ è un polinomio di grado d a coefficiente direttore positivo, allora si dimostra facilmente che $f = O(n^d)$. Analogamente, se $f = O(g)$ allora

$f+g = O(g)$ (cioè nello stimare una funzione possiamo trascurare i termini che crescono più lentamente di altri).

5. Si ha che $\log n = O(n^d)$, $n^d = O(e^n)$ per ogni d reale positivo, $n^2 \log n = O(n^{2+\epsilon})$ per ogni ϵ reale positivo.

Parlando alla buona, la relazione $f(n) = O(n^d)$ dice che la funzione f cresce all'incirca come la d-esima potenza della variabile. Ad esempio, se $n = 3$ ci dice che raddoppiare la variabile ha l'effetto di aumentare la funzione all'incirca di un fattore 8. La relazione $f(n) = O(\log^d n)$ dice che la funzione f cresce all'incirca come la d-esima potenza del numero di cifre binarie di n (questo perchè, a meno di una costante, il numero di cifre binarie di n è all'incirca $\log n$).

Si noti che la relazione $f(n) = O(1)$ significa che la funzione è limitata da una costante.

Se $f(n)$ indica il numero di cifre binarie di n, allora si ha che $f(n) = O(\log n)$. Se b è un'altra base fissata e $f(n)$ indica il numero di cifre di n in base b, si ha comunque $f(n) = O(\log n)$ (perchè in tal caso $\frac{1}{\log b}$ è una costante); se invece si permette che anche la base b aumenti, si ha che la funzione di 2 variabili $f(n, b)$ che indica il numero di cifre di n in base b è tale che $f(n, b) = O\left(\frac{\log n}{\log b}\right)$.

Per quanto abbiamo visto sull'addizione e la moltiplicazione di due interi (nel sistema binario o in un'altra base fissata) si ha che:

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

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

ove per "Tempo si intende tempo di calcolo necessario utilzzando un ben preciso algoritmo (quello sopra descritto).

In particolare

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

Se vogliamo stimare il tempo dell'addizione e della moltiplicazione in funzione degli interi m ed n anzichè del numero delle loro cifre, si ha

$$\text{Tempo}(m + n) = O(\max(\log m, \log n)),$$

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

Osserviamo inoltre che sono stati messi a punto algoritmi per la moltiplicazione molto più efficenti di quello da noi utilizzato (che è essenzialmente quello che si impara alla scuola elementare) e che permettono di moltiplicare due interi di k cifre in $O(k \log k \log(\log k))$ operazioni bit, che è meglio di $O(k^{1+\epsilon})$ per ogni ϵ.

Per quel che riguarda la sottrazione e la divisione (con resto) valgono le e seguenti stime:

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

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

Più precisamente, per trattare la sottrazione dobbiamo estendere la nostra definizione di operazione bit per includere l'operazione di sottrarre una cifra binaria da un'altra cifra binaria (eventualmente "prendendo a prestito una cifra 1 dalla colonna precedente), ed annotarne il risultato.

Per la divisione con resto di due numeri interi m ed n, di k ed l cifre binarie rispettivamente, si tratta semplicemente di continuare a sottrarre da m dei traslati di n, come nell'esempio seguente con $m = (11001001)_2$ ed $n = (100111)_2$, per cui vale la stessa stima della moltiplicazione. Si devono effettuare al più $k - l + 1$ sottrazioni di l operazioni bit ciascuna, per cui in tutto al più $k \cdot l$ operazioni bit (anche in questo caso non si tiene conto del tempo necessario per operazioni di tipo amministrativo, come quella di confrontare due interi, traslare un intero, ecc.).

1	1	0	0	1	0	0	1	:	1	0	0	1	1	1	=	1	0	1
1	0	0	1	1	1												
		1	0	1	1	0	1										
		1	0	0	1	1	1										
					1	1	0										

Concludiamo con una definizione di fondamentale importanza per la teoria degli algoritmi.

Definizione 1.2 *Un algoritmo per svolgere un calcolo a partire dagli interi $n_1, n_2, \ldots, n_r$, rispettivamente di $k_1, k_2, \ldots, k_r$ cifre binarie si dice un algoritmo "*a tempo polinomiale *se il numero di operazioni bit richieste è $O(k_1^{d_1} k_2^{d_2} \cdots k_r^{d_r})$ per opportuni interi $d_1, d_2, \ldots, d_r$.*

Quindi le operazioni elementari di somma, sottrazione, moltiplicazione e divisione con resto sono a tempo polinomiale.

Si dimostra che invece il calcolo di $n!$ non è a tempo polinomiale.

1.2 Divisibilità e fattorizzazione

Definizione 1.3 *Siano a e b due interi. Si dice che a* divide *b, in simboli $a|b$, se esiste un intero q tale che $b = qa$.*

Definizione 1.4 *Un intero p si dice* primo *se $p > 1$ e i suoi unici divisori positivi sono 1 e p stesso.*

Definizione 1.5 *Un intero n si dice* composto *se $n > 1$ e n non è primo.*

Ricordiamo alcune proprietà e teroremi importanti riguardo la divisibilità, i numeri primi, la fattorizzazione degli interi. Per le dimostrazioni si veda [7] o qualsiasi testo di algebra elementare.

Proprietà della divisibilità.

1. Se $a|b$ e c è un qualsiasi intero, allora $a|bc$.

2. Se $a|b$ e $b|c$, allora $a|c$.

3. Se $a|b$ e $a|c, alloraa|b \pm c$.

Teorema 1.1 (Euclide) *Esistono infiniti numeri primi.*

Una domanda che viene naturale porsi è come i primi siano distribuiti fra gli interi. Un famoso risultato congetturato nel diciottesimo secoli, ma provato solo alla fine del diciannovesimo da Jacques Hadamard (1865-1963) e Charles-Jean de la Vallè-Poussin (1865-1963) è il seguente:

Teorema 1.2 (Teorema dei numeri primi) *Sia $\pi(n)$ il numero di primi minori o uguali d n. Allora*

$$\lim_{n\to\infty} \frac{\pi(n)}{n/\log n} = 1,$$

cioè $\pi(n)$ tende asintoticamente a $\frac{n}{\log n}$.

Una conseguenza del Teorema dei numeri primi è che la densità media dei primi in un piccolo intervallo intorno ad x è circa $1/\log x$. Conoscere questa densità è importante perchè 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 quelle grandezza, e poi applichiamo un test di primalità a m, $m+1$, $m+2$, ...: ci aspetteremo di scoprire un primo al massimo dopo un numero di passaggi dell'ordine di $\log m$, che è 230 (pochi). (Naturalmente anzichè testare tutti gli interi a parture da m converrà testare solo quelli dispari, e la stima viene divisa per 2, oppure quelli congrui a ± 1 modulo 6, cioè non divisibili nè per 2 nè per 3, e la stima si abbassa ulteriormente...)

La seguente tabella mostra come la densità dei primi intorno ad x si approssimi bene con $1/\log x$. (Dati tratti da [11]).

intervallo da 10^n a $10^n + 1000$	numeri primi p	$1/\log(10^n)$
$10^6 < p < 10^6 + 1000$	75	0,0723...
$10^9 < p < 10^9 + 1000$	49	0,0482...
$10^{12} < p < 10^{12} + 1000$	37	0,0361...
$10^{15} < p < 10^{15} + 1000$	24	0,0289...

Teorema 1.3 (Teorema fondamentale dell'aritmetica) *Ogni intero n maggiore di 1 può essere scritto in modo unico (a meno dell'ordine dei fattori) come prodotto di numeri primi.*

Due conseguenze di questo teorema sono le seguenti:

1. Se un numero primo p divide ab, allora $p|a$ oppure $p|b$.

2. Se $m|a$ e $n|a$ e se m ed n non hanno divisori in comune maggiori di 1, allora $mn|a$.

Il teorema fondamentale dell'aritmetica dà anche un metodo sistematico per trovare tutti i divisori positivi di un intero positivo n, una volta che esso sia scritto come prodotto di primi. Infatti, se $n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_r^{\alpha_r}$, allora i divisori d di n sono tutti e soli quelli del tipo $d = p_1^{\beta_1} p_2^{\beta_2} \cdots p_r^{\beta_r}$, con $0 \leq \beta_i \leq \alpha_i$ per ogni $i = 1, \ldots, r$.

Definizione 1.6 *Siano a e b due interi non entrambi nulli. Si dice* massimo comun divisore di a e b, *denotato con* $\mathrm{MCD}(a, b)$, *il più grande intero positivo che divide sia a che b.*

Proposizione 1.4 *Siano a e b due interi non entrambi nulli. Allora* $\mathrm{MCD}(a, b)$ *è l'unico intero positivo d tale che:*

- *$d|a$ e $d|b$;*
- *se z è un intero tale che $z|a$ e $z|b$ allora $z|d$.*

Se si conosce la fattorizzazione in primi dei due interi a e b è facile calcolare il loro massimo comun divisore. Infatti basta prendere il prodotto di tutti i primi che compaiono in entrambe le fattorizzazioni, elevati al minore dei due esponenti con cui compaiono.

Terminiamo con altre due definizioni:

Definizione 1.7 *Si dice che due interi a e b sono* primi fra loro *se* $\mathrm{MCD}(a, b) = 1$.

Definizione 1.8 *Si dice che due intero n è composto se $n \neq \pm 1$ e n non è primo.*

1.3 L'algoritmo di Euclide

L'algoritmo di Euclide serve per trovare il massimo comun divisore di due interi positivi a e b, con $a > b$, senza dover fattorizzare a e b nel prodotto di primi. Lo ricordiamo per completezza.

Si divide prima a per b, ottenendo un quoziente q_1 ed un resto r_1: $a = bq_1 + r_1$. Poi si ripete il procedimento, con b ed r_1 al posto di a e b: $b = r_1 q_2 + r_2$. Procediamo ricorsivamente dividendo r_i per r_{i+1}, per $i \leq 1$, ottenendo $r_i = r_{i+1} q_{i+2} + r_{i+2}$, finchè otteniamo un resto nullo. L'ultimo resto non nullo r_l è $\mathrm{MCD}(a, b)$.

Proposizione 1.5 *L'algoritmo di Euclide fornisce il massimo comun divisore di due interi positivi $a > b$ in un numero finito di passi, in un tempo $O(\log^3 a)$.*

Dimostrazione. Sappiamo già che l'algoritmo termina (poichè $r_i < r_{i+1}$) e che l'ultimo resto non nullo è $\mathrm{MCD}(a,b)$ (si veda [7]).

Dimostriamo ora che i resti decrescono rapidamente, e precisamente $r_{i+2} < \frac{1}{2}r_i$. Infatti poichè i resti sono decrescenti; tutti i quozienti sono strettamente positivi, quindi

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

Ne segue che ci sono al più $2\lfloor \log_2 a \rfloor$ divisioni, e $2\lfloor \log_2 a \rfloor = O(\log a)$. Facciamo il conto in dettaglio:

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

ove r_l è l'ultimo resto non nullo e $\hat{l} = 2\left\lfloor \frac{l+1}{2} \right\rfloor - 1$, cioè $\hat{l} = l$ se l è dispari, $\hat{l} = l-1$ se l è pari; in ogni caso $\frac{l}{2} \leq \left\lfloor \frac{\hat{l}+1}{2} \right\rfloor \leq \lfloor \log_2 a \rfloor$.

Poichè ogni divisione coinvolge interi minori o uguali ad a, si esegue in $O(\log^2 a)$ operazioni bit. Quindi il tempo totale necessario è $O(\log a)O(\log^2 a) = O(\log^3 a)$. □

Teorema 1.6 (Identità di Bèzout) *Siano a e b due interi positivi, con $a > b$ e sia $d = \mathrm{MCD}(a,b)$. Allora esistono due interi u e v tali che*

$$d = au + bv.$$

Inoltre l'algoritmo per trovare u e v richiede $O(\log^3 a)$ operazioni bit.

Traccia della dimostrazione. Per trovare u e v è sufficiente ripercorrere le identità dell'algoritmo di Euclide, esprimendo ad ogni passo r_i come combinazione lineare di a e b a coefficienti interi. Si ha infatti $r_1 = a - bq_1$, $r_2 = b - r_1q_2 = -aq_2 + b(q_1 \cdot q_2 + 1)$, e in generale si sostituisce in $r_{i+2} = r_i - r_{i+1}q_{i+2}$ l'espressione di r_i e r_{i+1} come combinazione lineare di a e b. Più precisamente, si pongono $u_0 = 0$, $u_1 = 1$, $v_0 = 1$, $v_1 = q_1$ e, per convenzione, $r_0 = b$. Poi si definiscono ricorsivamente $u_{i+1} = u_{i-1} + u_iq_{i+1}$ e $v_{i+1} = v_{i-1} + v_iq_{i+1}$, per $i \geq 1$. Poi si dimostra per induzione che $r_i = (-1)^{i+1}u_ia + (-1)^i v_i b$ per ogni $i \geq 0$. Essendo $d = r_l$, si ottiene che $u = (-1)^{l+1}u_l$ e $v = (-1)^l v_l$.

È facile vedere che questo algoritmo richiede al più $O(\log^3 a)$ operazioni bit. □

L'algoritmo per trovare i due interi u e v come sopra si dice *algoritmo di Euclide esteso.*

Osservazione. È possibile migliorare la stima della complessità dell'algoritmo di Euclide, e di quello esteso, a $O(\log^2 a)$ anzichè $O(\log^3 a)$.

1.4 Le congruenze e l'anello $\mathbb{Z}/m\mathbb{Z}$

Definizione 1.9 *Dati due interi a e b, ed un intero positivo m si dice che "a* è congruo a b modulo m, *in simboli $a \equiv b \bmod m$, se $m|a-b$.*

Ricordiamo che la congruenza modulo m è una relazione di equivalenza e che l'insieme quoziente $\mathbb{Z}/m\mathbb{Z}$ delle classi resto modulo m è un anello, con le operazioni: $[a]+[b] = [a+b]$, $[a][b] = [ab]$, elemento neutro $[0]$ e unità $[1]$. Inoltre si ha che $\mathbb{Z}/m\mathbb{Z} = \{[0], [1], \ldots, [m-1]\}$.

Proposizione 1.7 *Una classe resto $[a]$ in $\mathbb{Z}/m\mathbb{Z}$ è invertibile se e solo se* $\mathrm{MCD}(a, m) = 1$. *Inoltre, se* $\mathrm{MCD}(a, m) = 1$ *(e $a < m$) l'inverso di $[a]$ può essere calcolato in $O(\log^3 m)$ operazioni bit.*

Dimostrazione. $[a]$ è invertibile se e solo se esiste un intero b tale che $ab \equiv 1 \mod m$. Supponiamo che $[a]$ sia invertibile. Allora $d = \mathrm{MCD}(a, m) | m | ab - 1$, inoltre $d|a$, quindi $d|1$, cioè $d = 1$. Viceversa, se $d = \mathrm{MCD}(a, m) = 1$ allora per il teorema 1.6 esistono u e v tali che $ua + vm = 1$. Prendendo $b = u$ si ha che $m|ua - 1 = ab - 1$. Osserviamo che non è restrittivo supporre $a < m$; in tal caso u e v sono calcolabili in $O(\log^3 m)$ operazioni bit, sempre per il teorema 1.6. □

Corollario 1.8 *Se p è un numero primo l'anello $\mathbb{Z}/p\mathbb{Z}$ è un campo.*

Dimostrazione. Segue immediatamente dalla proposizione 1.7. □

Definizione 1.10 *Siano a e b due interi, m un intero positivo ed x una indeterminata. La scrittura $ax \equiv b \mod m$ si dice* congruenza lineare *ed ogni intero z che sostituito ad x la rende vera si dice* soluzione della congruenza.

Ricordiamo il seguente teorema:

Teorema 1.9 (Teorema cinese dei resti) *Consideriamo il seguente sistema di congruenze lineari:*

$$\begin{aligned} x &\equiv a_1 \mod m_1 \\ x &\equiv a_2 \mod m_2 \\ &\cdots \quad \cdots \\ x &\equiv a_r \mod m_r \end{aligned}$$

ove $m_1, m_2, \ldots, m_r$ sono interi positivi maggiori di 1 a due a due primi fra loro (cioè $\mathrm{MCD}(m_i, m_j) = 1$ *per ogni $i \neq j$) e $a_1, a_2, \ldots, a_r$ sono interi qualunque; allora il sistema ammette una soluzione $z_0 \in \mathbb{Z}$, e ogni due soluzioni $z, z' \in \mathbb{Z}$ del sistema sono congrue modulo $M = m_1 m_2 \cdots m_r$.*

Dimostrazione. Poniamo $M_i = \frac{M}{m_i}$ per ogni $i = 1, \ldots r$. Essendo $\mathrm{MCD}(M_i, m_i) = 1$ esiste un intero N_i, calcolabile con l'algoritmo di Euclide, tale che $M_i N_i \equiv 1 \mod m_i$. D'altra parte si ha che $M_i N_i \equiv 0 \mod m_j$ se $i \neq j$. Ne segue che

$$z_0 = \sum_{i=0}^{r} a_i M_i N_i$$

è una soluzione del sistema.

Inoltre se $z, z' \in \mathbb{Z}$ sono due soluzioni del sistema, si ha che $z - z' \equiv 0 \mod m_i$ per ogni i, quindi $M|z - z'$. □

1.5 Algoritmi

Mostriamo ora l'implementazione dell'algoritmo di Euclide in Maple.

Ricordiamo che, dati due interi x e y, con $y \neq 0$, nella sintassi di Maple si ha che irem(x, y) è il resto della divisione di x per y.

Procedura per il calcolo del massimo comun divisore

```
mcd := proc(a,b)
local r, u, v;
if abs(a)>=abs(b) then
   u:=abs(a);
   v:=abs(b);
else v:=abs(a);
   u:=abs(b);
fi;
r:= irem(u,v);
while r>0 do
   u:=v;
   v:=r;
   r:= irem(u,v);
od;
v;
end;
```

Algoritmo di Euclide esteso

```
coef := proc(a,b)
local r, u, v,q,t1,t2,c0,c1,d0,d1,s,h;
if abs(a)>=abs(b) then
   u:=abs(a);
   v:=abs(b);
   else v:=abs(a);
   u:=abs(b);
fi;
r:= irem(u,v);
q:=iquo(u,v);
c0:=0;
c1:=1;
d0:=1;
d1:=q;
s:=1;
while r>0 do
   u:=v;
   v:=r;
   r:= irem(u,v);
   q:=iquo(u,v);
   t1:=c1;
```

```
   c1:=q*c1+c0;
   c0:=t1;
   t2:=d1;
   d1:=q*d1+d0;
   d0:=t2;
   s:=-s;
od; if abs(a)>abs(b) then h:=[-s*c0*sign(a),s*d0*sign(b)];
fi;
if abs(b)>abs(a) then h:=[s*d0*sign(a),-s*c0*sign(b)];
fi;
h;
end;
```

Capitolo 2

Generazione di numeri primi

Può essere utile avere a disposizione la lista di tutti i numeri primi minori o uguali ad un prefissato limite L. Infatti, per stabilire se un intero n è primo basterà scelgliere $L \geq n$ verificare se n appartiene o meno alla lista.

In questo capitolo esporremo tre metodi per generare una lista di numeri primi in ordine crescente: il metodo di divisione, il metodo di moltiplicazione e il crivello d Eratostene.

Osserviamo però che i metodi descritti sono efficienti solo per valori di L non troppo grandi, in quanto non sono a tempo polinomiale. Per verificare la primalità di interi n "grandi è quindi opportuno utilizzare altri metodi.

2.1 Metodo di divisione

Tale metodo si basa sulla seguente

Proposizione 2.1 *Sia $n > 1$ n intero. Se n non ha fattori primi p tali che $p \leq \sqrt{n}$, allora n è primo. Equivalentemente: se $n > 1$ è composto, allora ha un fattore primo p tale che $p \leq \sqrt{n}$.*

Dimostrazione. Dimostriamo la seconda delle due asserzioni equivalenti. Sia $n > 1$ composto, $n = ab$, con $a > 1$ e $b > 1$. Segue che a oppure b devono essere $\leq \sqrt{n}$, perchè se ambedue fossero $> \sqrt{n}$ dovrebbe essere $ab > n$. Supponiamo che sia $1 < a \leq \sqrt{n}$. Allora a ha un fattore primo p tale che $p \leq a$ e quindi $p \leq \sqrt{n}$. □

Si noti che un tale test richiede un numero di divisioni che non è polinomiale in n.

La proposizione 2.1 ci fornisce un primo elementare:

Test di primalità per divisioni successive *Per verificare se un intero $n > 1$ è primo basta verificare la sua non divisibilità per tutti i numeri p tali che $p \leq \sqrt{n}$.*

Per generare la lista dei numeri primi minori o uguali al L si può ora procedere applicando ad ogni intero dell'insieme $I = \{z \in \mathbb{Z} : 1 < z \leq L\}$ il test di primalità ora descritto (in realtà è sufficiente applicare i test ai numeri dispari, in quanto i numeri pari maggiori di 2 sono composti).

2.2 Metodo di moltiplicazione

Tale metodo è particolarmente adatto ad elencare i numeri primi p contenuti in un intervallo $I = \{z \in \mathbb{Z} : s < z \leq s + 2n - 1\}$, con $s, n \in \mathbb{Z}$, $s > 0$, $n > 0$, s pari.

Come si è detto in precedenza, il primo passo consiste nell'eliminare da I tutti i numeri pari. A questo punto l'insieme entro il quale si ricercano i numeri primi può essere rappresentato come $J = \{z \in \mathbb{Z} : z = s + 2k - 1, 1 \leq k \leq n\}$ ed numeri composti dovranno essere della forma $i \cdot j$, con $i > 1$, $j > 1$ ed ambedue dispari.

Sia $c = s + 2n - 1$ il massimo dell'insieme J. Per la proposizione 2.1 si ha che se $z \in J$ è composto, allora deve avere un fattore non banale i, con i dispari, $3 \leq i \leq \sqrt{c}$ e $j \geq \frac{s+1}{i}$. D'altra parte per il lemma 2.2 seguente, si ha che il più piccolo intero dispari maggiore o uguale a $\frac{s+1}{i}$ è $t = 2\left\lfloor \frac{s+1}{2i} \right\rfloor + 1$, quindi $t \leq j \leq \frac{c}{i}$.

Il metodo di moltiplicazione si descrive ora facilmente: basta calcolare tutti i prodotti del tipo $i \cdot j$, con $3 \leq i \leq \sqrt{c}$ e $t \leq j \leq \frac{c}{i}$ (avendo l'accortezza di porre $t = 3$ se $\left\lfloor \frac{s+1}{2i} \right\rfloor = 0$) e cancellarli dalla lista degli elementi di J.

Resta solo da dimostrare il seguente:

Lemma 2.2 *Siano s ed i interi, con $i > 0$. Allora il più piccolo intero dispari maggiore o uguale a $\frac{s+1}{i}$ è $2\left\lfloor \frac{s+1}{2i} \right\rfloor + 1$.*

Dimostrazione. Sia $x = \frac{s+i}{2i}$. Si ha che

$$\frac{s+1}{i} = 2x - 1 + \frac{1}{i}.$$

Poichè $\lfloor x \rfloor \leq x$ segue che

$$2\lfloor x \rfloor - 1 + \frac{1}{i} \leq \frac{s+1}{i},$$

e quindi

$$2\lfloor x \rfloor - 1 < \frac{s+1}{i}. \tag{2.1}$$

Sia ora $s = 2i \cdot q + r$, con $0 \leq r < 2i$.
Mostriamo che

$$\frac{s+1}{2i} \leq \lfloor x \rfloor + \frac{1}{2} \tag{2.2}$$

Questo equivale a mostrare che

$$q + \frac{r+1}{2i} \leq \left\lfloor q + \frac{s+i}{2i} \right\rfloor + \frac{1}{2}$$

Se $r < i$ abbiamo che $\frac{r+i}{2i} < 1$, quindi

$$\left\lfloor q + \frac{r+i}{2i} \right\rfloor = q.$$

D'altra parte il primo membro della disuguaglianza 2.2 è

$$q + \frac{r+1}{2i} \leq q + \frac{i}{2i} = q + \frac{1}{2},$$

quindi 2.2 è vera.
Se invece $i \leq r < 2i$ si ha che $1 \leq \frac{r+i}{2i} \leq \frac{3}{2}$, quindi

$$\left\lfloor q + \frac{r+i}{2i} \right\rfloor = q + 1,$$

mentre il primo membro della disuguaglianza 2.2 è

$$q + \frac{r+1}{2i} \leq q + \frac{3i}{2i} = q + \frac{3}{2},$$

e anche in questo caso 2.2 è vera.
Dall'equazione 2.2, moltiplicando entrambi i membri per 2, e dall'equazione 2.1, segue che:

$$2\lfloor x \rfloor - 1 < \frac{s+1}{i} \leq 2\lfloor x \rfloor + 1,$$

come volevasi dimostrare. □

2.3 Il crivello di Eratostene

Il seguente metodo per determinare tutti i numeri primi minori o uguali ad un intero positivo n è attribuito ad Eratostene, matematico di Cirene che visse tra il 256 e il 194 A.C. circa.

Anch'esso consiste nell'elimiare dalla lista $I = \{z \in \mathbb{Z} : 1 < z \leq n\}$ tutti i numeri composti.

Si procede come segue. Il primo intero (vale a dire 2) è primo. Sia $p_1 = 2$. Il primo passo consiste nell'eliminare da I tutti i multipli di 2. Il primo numero > 2 non eliminato è $p_2 = 3$, che è primo, e al secondo passo eliminiamo anche tutti i multipli di 3. Ora il primo numero non eliminato maggiore di 3 è $p_3 = 5$ e si eliminano tutti i suoi multipli. In generale, al k-esimo passo si considera il primo numero p_k non eliminato maggiore di p_{k-1}, e si eliminano tutti i suoi multipli.

Ad un certo punto il procedimento termina (perchè la successione $p_1 > p_2 > \cdots > p_k > \cdots$ è strettamente crescente).

Dimostriamo ora che i numeri non eliminati sono tutti e soli i numeri primi minori o uguali ad n.

Dimostriamo per induzione che dopo il k-esimo passo i primi $k+1$ numeri rimasti $p_1 = 2, p_2 = 3, \ldots, p_{k+1}$ sono primi consecutivi. Questo è sicuramente vero per $k = 1$ perchè 2 e 3 sono primi consecutivi. Al $k+1$-esimo passo vengono eliminati tutti i multipli di p_{k+1} ad eccezione di p_{k+1} stesso. Il numero primo q successivo a p_{k+1} non sarà sicuramente stato eliminato (non essendo multiplo di nessun primo ad esso precedente), ma tutti i numeri compresi trea p_{k+1} e q, che sono composti, saranno stati eliminati, in quanto sono divisibili per qualche primo minore di q. Il primo numero maggiore di p_{k+1} rimasto, cioè p_{k+2} è quindi proprio q, il numero primo successivo a p_{k+1}, come volevasi dimostrare.□

Osserviamo che l'algoritmo può essere reso più veloce con qualche semplice accorgimento. Quando dobbiamo eliminare i multipli di un certo primo $p > 2$, i numeri $2p, 3p, \ldots, (p-1)p$ saranno già stati eliminati in qualche passo precedente. Quindi il primo numero da eliminare sarà p^2. Inoltre i multipli di p del tipo $p^2 + sp$, con s dispari, saranno già stati eliminati perchè sono multipli di 2. Quindi possiamo iniziare il processo da eliminazione dei multipli di p a partire da p^2 e procedendo a passi di $2p$, cioè eliminare $p^2 + 2p$, poi $p^2 + 4p$, $p^2 + 6p$ e così via. Una volta arrivati ad un primo $p_k \geq \sqrt{n}$, ci si può fermare.

La seguente procedura in Maple usa il crivello di Eratostene per determinare la lista di tutti i primi minori o uguali ad n. In questo algoritmo, anzichè partire dalla lista $I = \{z \in \mathbb{Z} : 1 < z \leq n\}$, si considerano solo i numeri dispari, cioè si parte dalla lista $J = \{2i + 1 \in \mathbb{Z} : 1 \leq i \leq \lfloor \frac{n-1}{2} \rfloor\}$.

Si parte da una lista H con $\frac{n-1}{2}$ elementi, e ciascuno di essi ha il valore $true$. Ogni volta che si vuole cancellare un numero del tipo $2i+1$, si assegna il valore $false$ all'i-esimo elemento della lista.

```
era := proc(n)
local i, j, k,l, H, L;
H:=[seq(true,x=1..floor((n-1)/2))];
j:=1;
while (2*j+1)^2 < n do
  i:=((2*j+1)^2-1)/2;
  while i<= floor((n-1)/2) do
    H:=subsop(i=false,H);
    i:=i+2*j+1;
  od;
  k:=j+1;
  while H[k]=false do
    k:=k+1;
  od;
  j:=k;
od;
L:=[2];
```

```
for l from 1 to floor((n-1)/2) do
   if H[l]=true then L := [op(L), 2*l+1];
   fi;
od;
L;
end;
```

Capitolo 3

Il teorema di Fermat

3.1 La funzione φ di Eulero

Definizione 3.1 *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 o uguali ad n che sono coprimi con n.*

Si noti che, equivalentemente, per la proposizione 1.7 $\varphi(n)$ è l'ordine del gruppo moltiplicativo 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

$$\varphi(p^\alpha) = p^\alpha - p^{\alpha-1} = p^{\alpha-1}(p-1) = p^\alpha \left(1 - \frac{1}{p}\right). \tag{3.1}$$

(Questo perchè gli interi primi con p^α sono quelli che non sono multipli di p).

Proposizione 3.1 *Siano m ed n due interi maggiori di 1 tali che $\mathrm{MCD}(m,n) = 1$ allora l'anello $\mathbb{Z}/mn\mathbb{Z}$ è isomorfo al prodorro diretto $\mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$.*

Dimostrazione. Si noti che l'omomorfismo di anelli

$$f : \mathbb{Z} \to \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$$
$$z \mapsto ([z]_m, [z]_n)$$

ha per nucleo l'insieme degli interi che sono divisibili sia per m sia per n, e poichè $\mathrm{MCD}(m,n) = 1$ questo è l'insieme dei multipli di mn. Inoltre f è suriettivo per il teorema cinese dei resti 1.9, quindi per il primo teorema di omomorfismo di anelli si ha che $\mathbb{Z}/mn\mathbb{Z}$ è isomorfo a $\mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$. □

Se A è un anello, denoteremo con A^* il gruppo moltiplicativo degli elementi invertibili di A.

Proposizione 3.2 *Siano A e B due anelli. Si ha che il gruppo moltiplicativo $(A \times B)^*$ è isomorfo al prodotto diretto $A^* \times B^*$.*

Dimostrazione. Basta osservare che gli elementi invertibili di $A \times B$ sono tutti e soli quelli del tipo (a, b), ove a è un elemento invertibile di A e b è un elemento invertibile di B.

Proposizione 3.3 *La funzione φ di Eulero è* **moltiplicativa**, *cioè, dati due interi m ed n maggiori di 1 tali che* $\mathrm{MCD}(m, n) = 1$, *si ha che*

$$\varphi(mn) = 1 \qquad \text{se } \mathrm{MCD}(m, n) = 1$$

Dimostrazione. Per la proposizione 3.1 si ha che $\mathbb{Z}/mn\mathbb{Z}$ è isomorfo a $\mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$. Inoltre dalla proposizione 3.2 segue che il numero degli elementi invertibili $\varphi(mn)$ del primo anello è uguale a quello del secondo, che è $\varphi(m)\varphi(n)$. □

Come corollario otteniamo una formula per il calcolo di $\varphi(n)$.

Corollario 3.4 *Sia $n > 1$ un numero intero e sia $n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_r^{\alpha_r}$ la sua fattorizzazione in primi. Allora*

$$\varphi(n) = \prod_{i=1}^{r} (p_i^{\alpha_i} - p_i^{\alpha_i - 1}) = n \prod_{i=1}^{r} \left(1 - \frac{1}{p_i}\right)$$

Dimostrazione. Segue direttamente dalla proposizione 3.3 e dall'uguaglianza 3.1. □

Proposizione 3.5 *Per ogni intero $n \geq 1$ si ha:*

$$n = \sum_{d|n} \varphi(d),$$

ove la somma è estesa a tutti i divisori di n.

Dimostrazione. Sia $\mathcal{C} = \{x \in \mathbb{Z} : 1 \leq x \leq n\}$. Per ogni divisore d di n sia $\mathcal{C}_d = \{x \in \mathbb{Z} : d \leq x \leq n \text{ e } \mathrm{MCD}(x, n) = d\}$. Gli insiemi $\mathcal{C}_d$, al variare di d, costituiscono una partizione di $\mathcal{C}$. Infatti, dati due divisori distinti d e d' di n, gli insiemi $\mathcal{C}_d$ e $\mathcal{C}_{d'}$ sono disgiunti ed ogni elemento di $\mathcal{C}$ appartiene a qualche $\mathcal{C}_d$. Segue che $n = |\mathcal{C}| = \sum_{d|n} |\mathcal{C}_d|$ (ove $|\mathcal{S}|$ indica il numero di elementi dell'insieme $\mathcal{S}$). Se $x \in \mathcal{C}_d$, allora d divide x, $1 \leq \frac{x}{d} \leq \frac{n}{d}$ e $\mathrm{MCD}(\frac{x}{d}, \frac{n}{d}) = 1$. Quindi l'insieme $\mathcal{C}_d$ è in corrispondenza biunivoca con l'insieme degli interi y tali che $1 \leq y \leq \frac{n}{d}$ e $\mathrm{MCD}(y, \frac{n}{d}) = 1$, cosicchè $|\mathcal{C}_d| = \varphi(\frac{n}{d})$. Poichè anche $\frac{n}{d}$ assume tutti i valori possibili dei divisori di n si ha che: $n = \sum_{d|n} \varphi(\frac{n}{d}) = \sum_{d|n} \varphi(d)$. □

3.2 I teoremi di Fermat, Eulero e Wilson

Il seguente teorema fu scoperto da Fermat nel 1640, ma come per molti altri suoi risultati, non ci è pervenuta alcuna sua dimostrazione. Fu Eulero nel 1736 a pubblicarne la prima dimostrazione.

Teorema 3.6 *Siano p un primo e a un intero qualsiasi. Allora $a^p \equiv a \ mod \ p$. Se p non divide a, allora $a^{p-1} \equiv 1 \ mod \ p$.*

Prima di domostrarlo, ricordiamo il seguente teorema sui gruppi finiti (si veda [7]):

Teorema 3.7 (Teorema di Lagrange) *Sia $(G, \cdot, e)$ un gruppo finito di ordine $|G| = n$, H un suo sottogruppo di ordine $|H| = m$ e g un elemento di G. Allora $m|n$ e $g^n = e$.*

Dimostrazione del teorema 3.6. Supponiamo dapprima che $p \nmid a$. Poichè p è primo, per il corollario 1.8 $\mathbb{Z}/pZ$ è un campo, quindi il gruppo moltiplicativo degli elementi invertibili $(\mathbb{Z}/pZ)^*$ ha ordine $p-1$. Per il teorema 3.7 si ha quindi che $[a]^{p-1} = [1]$ in $\mathbb{Z}/pZ$, da cui la tesi. Moltiplicando per a si ottiene $a^p \equiv a$ mod p. L'ultima uguaglianza è vera anche se $p|a$, infatti in tal caso entrambi i membri sono congrui a 0 modulo p.□

Il seguente teorema è una generalizzazione del teorema di Fermat.

Teorema 3.8 (Teorema di Eulero) *Sia $m \geq 2$ un numero intero e sia a un intero coprimo con m. Allora $a^{\varphi(m)} \equiv 1 \ mod \ m$.*

Dimostrazione. Anche questo teorema segue dal teorema 3.7, osservando che $[a]$ è invertibile in $\mathbb{Z}/m\mathbb{Z}$ per la proposizione 1.7 e che il gruppo moltiplicativo degli elementi invertibili di $\mathbb{Z}/m\mathbb{Z}$ ha ordine $\varphi(m)$.□

Il teorema di Wilson fornisce in realtà un test di primalità. Il suo interesse è però puramente teorico, in quanto la sua implementazione algoritmica richiede di calcolare $(n-1)!$.

Teorema 3.9 (Teorema di Wilson) *Un intero $p > 1$ è primo se e solo se $(p-1)! \equiv -1 \ mod \ p$.*

Dimostrazione. Supponiamo che p sia primo. Se $p = 2$ il risultato è vero. Sia dunque $p > 2$. Poichè per il corollario 1.8 $\mathbb{Z}/pZ$ è un campo, gli unici elementi invertibili di $\mathbb{Z}/pZ$ che sono inversi di se stessi sono $[1]$ e $[p-1] = [-1]$, in quanto sono radici del polinomio $x^2 - [1]$. Gli altri elementi hanno inverso nell'insieme $\{[2], \ldots, [p-2]\}$ cosicchè in $\mathbb{Z}/pZ$ si ha che il prodotto $[2] \cdots [p-2]$ può essere scritto nella forma $[x_1][x_1]^{-1}[x_2][x_2]^{-1} \cdots [x_{(p-3)/2}][x_{(p-3)/2}]^{-1}$ e dunque $[(p-1)!] = [-1]$, da cui la tesi.

Viceversa, se $p = ab$ è composto, con $a > 1$ e $b > 1$ e per assurdo $p|(p-1)!+1$ si ha che $a|(p-1)!$, poichè $1 < a < p$, inoltre $a|p$, quindi $a|1$, contro l'ipotesi $a > 1$.□

3.3 Un algoritmo per il calcolo delle potenze modulo m

Per poter utilizzare in pratica il teorema di Fermat (ad esempio per dimostrare che un mumero intero p non è primo esibendo un intero a tale che $p \nmid a$ e

$a^{p-1} \not\equiv 1 \bmod p$) è necessario essere in grado di calcolare efficientemente una potenza a^n modulo m.

Il metodo peggiore possibile è quello di calcolare a^n e poi ridurre modulo m, in quanto a^n risulterà probabilmente un numero molto grande.

Una prima accortezza è quella di ridurre modulo m ogni volta che si sia eseguita una moltiplicazione. Più precisamente, se $\langle x \rangle$ indica il resto della divisione di x per m (cioè $\langle x \rangle$ è la riduzione di x modulo m), si calcola $\langle a \rangle$ e lo si moltiplica per a, poi si riduce modulo m ottenedo $\langle a^2 \rangle = \langle a \cdot \langle a \rangle \rangle$ e così via. (Si noti che siamo utilizzando il fatto che se $a^k \equiv b \bmod m$ allora $a^{k+1} \equiv ab \bmod m$). Si dimostra che un tale algoritmo richiede $O(n \cdot \log^2 m)$ operazioni bit, contro le $O(n^2 \cdot \log^2 m)$ operazioni bit dell'algoritmo precedente.

Una drastica riduzione del tempo si ottiene con il metodo dei quadrati ripetuti, che è a tempo plonomiale (cioè polinomiale in $\log n$ e $\log m$), in quanto richiede $O(\log n \cdot \log^2 m)$ operazioni bit.

Si procede a partire dalla notazione binaria di $n = (d_{k-1} d_{k-2} \cdots d_0)$, per cui sarà

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

Si calcolano tutte le potenze del tipo b^{2^i} modulo m per $i = 0, 1, \ldots, k-1$, ottenendo ciascuna come quadrato della precedente e riducendo subito modulo m. Ciascun passo richiede $O(\log^2 m)$ operazioni bit quindi vengono eseguite in tutto $O(\log n \cdot \log^2 m)$ operazioni bit. Poi vengono moltiplicate tra loro le potenze $\overline{b^{2^i}}$ corrispondenti agli $n_i = 1$ (di nuovo riducendo modulo m dopo ogni moltiplicazione), e questo passo richiede ancora $O(\log n \cdot \log^2 m)$ operazioni bit.

A questo punto citiamo un algoritmo particolarmente efficiente per calcolare i prodotti modulo m, l'*algoritmo di Head*. Per una sua descrizione si veda [3, §3.3].

Riportiamo ora la procedura in Maple per il calcolo della potenza a^m modulo n.

```
pow :=proc(a,m,n)
local A, p, r,q;
A:=modp(a,n);
p:=1;
q:=m;
while q>0 do
  r:=modp(q,2);
  if r=1 then p:=modp(p*A,n);
  fi;
  A:=modp(A^2,n);
  q:=(q-r)/2;
od;
p;
end;
```

Capitolo 4

La struttura di $(\mathbb{Z}/mZ)^*$ e le radici primitive

4.1 La struttura di $(\mathbb{Z}/pZ)^*$

Per dimostrare la prossima importante proposizione, ricordiamo prima un lemma sui gruppi.

Lemma 4.1 *Sia $(G, \cdot, e)$ un gruppo commutativo, e siano b_1, b_2 due suoi elementi di ordine n_1 ed n_2 rispettivamente, con $\mathrm{MCD}(n_1, n_2) = 1$. Allora l'elemento $b = b_1 b_2$ ha per ordine $n_1 n_2$.*

Dimostrazione. Sia r l'ordine di b. Si ha che $b^{n_1 n_2} = b_1^{n_1 n_2} b_2^{n_1 n_2} = e$, quindi $r | n_1 n_2$. Si ha inoltre che $e = (b^r)^{n_1} = b^{n_1 r} = (b_1)^{n_1 r}(b_2)^{n_1 r} = (b_1{}_1^n)^r (b_2)^{n_1 r} = (b_2)^{n_1 r}$, quindi $n_2 | n_1 r$ e poichè $\mathrm{MCD}(n_1, n_2) = 1$ segue che $n_2 | r$. Analogamente si ottiene che $n_1 | r$, quindi $n_1 n_2 | r$ (utilizzando ancora una volta il fatto che $\mathrm{MCD}(n_1, n_2) = 1$). Segue che $r = n_1 n_2$, come volevasi. □

Proposizione 4.2 *Sia G un sottogruppo finito del gruppo moltiplicativo di un campo. Allora G è ciclico.*

Dimostrazione. Sia $|G| = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_r^{\alpha_r}$ la fattorizzazione delll'ordine di G, ove i p_i sono primi distinti. Il polinomio $x^{|G|/p_i} - 1$ ha al più $|G|/p_i$ radici, esiste $a_i \in G$ tale che $a_i^{|G|/p_i} \neq 1$. Sia ora $b_i = a_i^{|G|/p_i^{\alpha_i}}$. Si ha che $b_i^{p_i^{\alpha_i - 1}} = a_i^{|G|/p_i} \neq 1$ e $b_i^{p_i^{\alpha_i}} = 1$, quindi b_i ha ordine p^{α_i}. Sia ora $b = b_1 b_2 \cdot b_r$. Poichè i b_i commutano tra loro e i loro ordini sono a due a due coprimi, l'ordine di b è il prodotto degli ordini dei b_i, cioè n (segue dal lemma precedente). Poichè il gruppo ciclico generato da b è un sottogruppo di G di ordine n, deve coincidere con G. □

Corollario 4.3 *Sia p un numero primo, allora il sottogruppo moltiplicativo $(\mathbb{Z}/pZ)^*$ di $\mathbb{Z}/pZ$ è ciclico.*

Dimostrazione. Segue dalla proposizione 4.2 e dal fatto che $\mathbb{Z}/pZ$ è un campo (per il corollario 1.8). □

4.2 Radici primitive

Sia $m > 1$ un intero e sia a un intero tale che $\mathrm{MCD}(a, m) = 1$. Allora per la proposizione 1.7 $[a] \in (\mathbb{Z}/mZ)^*$. Inoltre per il teorema di Lagrange 3.7 si ha che l'ordine di $[a]$ divide l'ordine di $(\mathbb{Z}/mZ)^*$, cioè $\varphi(n)$. È interessante il caso in cui $[a]$ abbia ordine esattamente $\varphi(m)$.

Definizione 4.1 *Sia $m > 1$ un intero. Un intero a si dice una* radice primitiva modulo m *se* $\mathrm{MCD}(a, m) = 1$ *e $[a]$ ha ordine $\varphi(m)$ in $(\mathbb{Z}/pZ)^*$.*

Osservazione. Non è detto che, dato m, esista una radice primitiva a modulo m; se però un tale a esiste, segue immediatamente che $(\mathbb{Z}/mZ)^*$ è un gruppo ciclico, generato da $[a]$. Inoltre, se $m = p$ è primo, allora segue da corollario 4.3 che esiste una radice primitiva modulo p.

Proposizione 4.4 *Sia p un numero primo, e $\alpha \geq 1$ un intero. Si consideri la seguente applicazione*

$$f : (\mathbb{Z}/p^{\alpha+1}\mathbb{Z})^* \to (\mathbb{Z}/p^{\alpha}\mathbb{Z})^*$$

$$[x]_{p^{\alpha+1}} \mapsto [x]_{p^{\alpha}}.$$

Si ha che f è un omomorfismo di gruppi suriettivo il cui nucleo è

$$\mathrm{Ker}(f) = \{[1 + ap^{\alpha}]_{p^{\alpha+1}} : a = 0, 1, \ldots p - 1\}.$$

Inoltre $\mathrm{Ker}(f)$ *ha ordine p.*

Dimostrazione. f è la restrizione a $(\mathbb{Z}/p^{\alpha+1}\mathbb{Z})^*$ dell'omomorfismo di gruppi $g : \mathbb{Z}/p^{\alpha+1}\mathbb{Z} \to \mathbb{Z}/p^{\alpha}\mathbb{Z}$ definito da $g([x]_{p^{\alpha+1}}) = [x]_{p^{\alpha}}$ (si noti che g è ben definito). Poichè se $\mathrm{MCD}(x, p^{\alpha}) = 1$ si ha anche $\mathrm{MCD}(x, p^{\alpha+1}) = 1$, f è suriettivo. Si ha che $\mathrm{Ker}(f) = \{[1 + ap^{\alpha}]_{p^{\alpha+1}} : a \in \mathbb{Z}\}$, ma è chiaro che quello che conta è il resto della divistone di a per p, infatti se $a = p \cdot q + r$, con $0 \leq r < a$, si ha che $[1 + ap^{\alpha}]_{p^{\alpha+1}} = [1 + rp^{\alpha}]_{p^{\alpha+1}}$. Inoltre per $a = 0, 1, \ldots p - 1$ le classi $[1 + ap^{\alpha}]_{p^{\alpha+1}}$ sono tutte distinte, quindi $\mathrm{Ker}(f)$ ha ordine p. □

Lemma 4.5 *Siano G ed H due gruppi ed $f : G \to H$ un omomorfismo suriettivo. Sia $S \subseteq G$ un sottogruppo tale che $f(S) = H$ e* $\mathrm{Ker}(f) \subseteq S$. *Allora $S = G$.*

Dimostrazione. Dimostriamo che ogni elemento x di G appartiene a S. Sia $x \in G$. Poichè $H = f(G) = f(S)$ esiste $s \in S$ tale che $f(x) = f(s)$. Inoltre $xs^{-1} \in \mathrm{Ker}(f) \subseteq S$, quindi $x = (xs^{-1})s \in S$. □

Proposizione 4.6 *Sia p un numero primo.*

a) Un intero x è una radice primitiva modulo p^2 se soddisfa le seguenti condizioni:

i) x è una radice primitiva modulo p,
ii) $x^{p-1} \not\equiv 1 \ mod\ p^2$.

b) Se z è una radice primitiva modulo p, ma non una radice primitiva modulo p^2, allora $z+p$ è una radice primitiva modulo p^2.

Dimostrazione. a) Sia x è una radice primitiva modulo p tale che $x^{p-1} \not\equiv 1 \mod p^2$. Dimostriamo che $[x]_{p^2}$ genera $(\mathbb{Z}/p^2\mathbb{Z})^*$. Sia f l'omomrfismo

$$f : (\mathbb{Z}/p^2\mathbb{Z})^* \to (\mathbb{Z}/p\mathbb{Z})^*$$

della proposizione 4.4. Utilizziamo il lemma 4.5, ove $G = (\mathbb{Z}/p^2\mathbb{Z})^*$, $H = (\mathbb{Z}/p\mathbb{Z})^*$ e S è il sottogruppo generato da $[x]_{p^2}$. Poichè x è una radice primitiva modulo p, si ha che $[x]_p$ genera $(\mathbb{Z}/p\mathbb{Z})^*$, quindi $f(S) = (\mathbb{Z}/p\mathbb{Z})^*$. Inoltre $[x]_{p^2}^{p-1} \in \ker(f)$, e si ha che $[x]_{p^2}^{p-1} \neq [1]_{p^2}$ perchè $x^{p-1} \not\equiv 1 \mod p^2$, quindi $[x]_{p^2}^{p-1}$ genera $\ker(f)$ (si ricordi che $\ker(f)$ ha ordine p per la proposizione 4.4). Per il lemma 4.5 segue che $S = (\mathbb{Z}/p^2\mathbb{Z})^*$, come volevasi dimostrare.
b) Sia ora z una radice primitiva modulo p, ma non una radice primitiva modulo p^2. Si ha che $z^{p-1} \equiv 1 \mod p^2$. Chiaramente $z+p$ è ancora una radice primitiva modulo p, essendo $[z]_p = [z+p]_p$. Per la parte a) basta dimostrare che $(z+p)^{p-1} \not\equiv 1 \mod p^2$. Si ha che

$$(z+p)^{p-1} = z^{p-1} + \binom{p-1}{1} z^{p-2}p + \binom{p-1}{2} z^{p-3}p^2 + \cdots + p^{p-1},$$

quindi

$$(z+p)^{p-1} \equiv z^{p-1} + (p-1)z^{p-2}p \equiv 1 - z^{p-2}p \qquad \mod\ p^2.$$

Ora poichè $z^{p-2}p \not\equiv 0 \mod p^2$, segue che $(z+p)^{p-1} \not\equiv 1 \mod p^2$, come volevasi dimostrare. □

Corollario 4.7 *Se p è un numero primo, il gruppo moltiplicativo $(\mathbb{Z}/p^2\mathbb{Z})^*$ è ciclico.*

Dimostrazione. Per il corollario 1.8 esiste una radice primitiva x modulo p. Allora per la proposizione 4.6 si ha che x oppure $x+p$ è una radice primitiva modulo p^2. □

Proposizione 4.8 *Sia p un primo dispari. Se x è una radice primitiva modulo p^2, allora x è anche una radice primitiva modulo p^α.*

Dimostrazione. Dimostriamo per induzione su $k \geq 2$ che: se x è una radice primitiva modulo p^k, allora x è anche una radice primitiva modulo p^{k+1}.
Si procede esattamente come per la dimostrazione della proposizione 4.6 *a)*, considerando l'omomorfismo

$$f : (\mathbb{Z}/p^{k+1}\mathbb{Z})^* \to (\mathbb{Z}/p^k\mathbb{Z})^*,$$

della proposizione 4.4 e utilizzando il lemma 4.5, ove $G = (\mathbb{Z}/p^{k+1}\mathbb{Z})^*$, $H = (\mathbb{Z}/p^k\mathbb{Z})^*$ e S è il sottogruppo generato da $[x]_{p^{k+1}}$. Quindi è sufficiente mostrare che $([x]_{p^{k+1}})^{p^{k-1}(p-1)} \in \ker(f)$ è un elemento diverso da $[1]_{p^{k+1}}$, cioè che:

$$x^{p^{k-1}(p-1)} \not\equiv 1 \qquad \mod\ p^{k+1}.$$

Poichè per ipotesi induttiva $[x]_{p^k}$ ha ordine $p^{k-1}(p-1)$, si ha che

$$x^{p^{k-2}(p-1)} \not\equiv 1 \qquad \mod\ p^k,$$

inoltre il teorema di Eulero 3.8 $x^{p^{k-2}(p-1)} \equiv 1 \mod p^{k-1}$, quindi $x^{p^{k-2}(p-1)} = 1 + ap^{k-1}$ con $p \nmid a$. Segue che

$$x^{p^{k-1}(p-1)} = (1 + ap^{k-1})^p \equiv 1 + ap^k \qquad \mod\ p^{k+1}.$$

Questo conclude la dimostrazione. □

Poichè per il corollario 1.8 esiste una radice primitiva x modulo p, la proposizione precedente dimostra che:

Teorema 4.9 *Se p è un primo dispari e $\alpha \geq 1$ è un intero, allora il gruppo moltiplicativo $(\mathbb{Z}/p^\alpha\mathbb{Z})^*$ è ciclico.*

Ci occupiamo ora del caso $p = 2$.

Teorema 4.10 *Sia $\alpha \geq 1$ un intero. Allora il gruppo moltiplicativo $(\mathbb{Z}/2^\alpha\mathbb{Z})^*$ è ciclico se e solo se $\alpha = 1$ oppure $\alpha = 2$.*

Dimostrazione. Si ha che $(\mathbb{Z}/2\mathbb{Z})^* = \{[1]_2\}$ è banalmente ciclico, come pure è ciclico $(\mathbb{Z}/2^2\mathbb{Z})^* = \{[1]_4, [-1]_4\}$. Sia quindi $\alpha \geq 3$. Ricordiamo che in un gruppo ciclico $(G, \cdot, e)$ c'è al più un elemento g diverso da e tale che $g^2 = e$, invece in $(\mathbb{Z}/2^\alpha\mathbb{Z})^*$ ve ne sono almeno 3: $[-1]_{2^\alpha}, [1 + 2^{\alpha-1}]_{2^\alpha}$ e $[-1 + 2^{\alpha-1}]_{2^\alpha}$. □

Citiamo a questo punto che per $\alpha \geq 3$ il gruppo $(\mathbb{Z}/2^\alpha\mathbb{Z})^*$ è il prodotto diretto di un gruppo ciclico di ordine 2 e un gruppo ciclico di ordine $2^{\alpha-2}$. Per una dimostrazione, si veda [8, Theorem 4.20].

Ricordiamo la seguente definizione:

Definizione 4.2 *Sia $(G, \cdot, e)$ un gruppo finito. L'*esponente *di G è il più piccolo intero positivo n tale che si abbia $g^n = e$ per ogni elemento g di G.*

In pratica l'esponente di un gruppo finito G è il minimo comune multiplo degli ordini degli elementi di G, e quindi per il teorema di Lagrange divide l'ordine di G. Inoltre, se G è un gruppo ciclico l'esponente di G è uguale all'ordine di G.

Lemma 4.11 *Sia m un intero tale che $m = r \cdot s$, con r ed s interi, $r > 2$, $s > 2$ e $\mathrm{MCD}(r, s) = 1$. Allora non esistono radici primitive modulo m.*

Dimostrazione. Si ha che $(\mathbb{Z}/m\mathbb{Z})^*$ è isomorfo a $(\mathbb{Z}/r\mathbb{Z})^* \times (\mathbb{Z}/s\mathbb{Z})^*$. Inoltre, poichè r ed s sono maggiori di 2, $\varphi(r)$ e $\varphi(s)$ sono entrambi interi pari. Sia $([x]_r, [y]_s) \in (\mathbb{Z}/r\mathbb{Z})^* \times (\mathbb{Z}/s\mathbb{Z})^*$. Si ha che

$$([x]_r, [y]_s)^{\frac{\varphi(r)\varphi(s)}{2}} = (([x]_r^{\varphi(r)})^{\frac{\varphi(s)}{2}}, ([y]_s^{\varphi(s)})^{\frac{\varphi(r)}{2}}) = ([1]_r, [1]_s),$$

quindi l'esponente di $(\mathbb{Z}/m\mathbb{Z})^*$ è minore o uguale a $\frac{\varphi(r)\varphi(s)}{2}$. Ma $(\mathbb{Z}/m\mathbb{Z})^*$ ha ordine $\varphi(m) = \varphi(r)\varphi(s)$, quindi non può essere ciclico. □

Possiamo ora determinare esattamente per quali interi m esistono radici primitive.

Teorema 4.12 *Sia $m > 1$ un intero. Allora esisiste una radice primitiva modulo m se e solo se m è del tipo: $m = 2$, $m = 4$,$m = p^k\alpha$ oppure $m = 2p^\alpha$, con p dispari.*

Dimostrazione. Se m è potenza di un primo si utilizzano i teoremi 4.9 e 4.10. Supponiamo dunque che m sia per almeno due primi distinti e che esista una radice primitiva modulo m. Per il lemma 4.11 si può supporre $m = 2p^\alpha$, con p dispari. Segue che $(\mathbb{Z}/m\mathbb{Z})^* \cong (\mathbb{Z}/2\mathbb{Z})^* \times (\mathbb{Z}/p^\alpha\mathbb{Z})^* \cong (\mathbb{Z}/p^\alpha\mathbb{Z})^*$ è ciclico. (Qui il simbolo "$\cong$" indica che due gruppi sono isomorfi). □

Osservazione. Sia $m > 1$ un intero. Se esiste una radice primitiva modulo m allora ne esistono esattamente $\varphi(\varphi(m))$, infatti le radici primitive modulo m, se esistono, sono esattamente i generatori del gruppo ciclico $(\mathbb{Z}/m\mathbb{Z})^*$ che ha ordine $\varphi(m)$.

Se p è un primo, il numero di radici primitive modulo p è dunque $p - 1$. Si dimostra che, anche se può variare considerevolmente, $\phi(n)$ è in media $\frac{6n}{\pi^2}$, all'incirca i 2/3 di n. Per gli interi del tipo $p-1$ tale stima è troppo ottimistica, in ogni caso è ragionevole aspettarsi che una porzione significativa di numeri minori di $p - 1$ sia costituita da radici primitive. Per trovarne una, è sensato scegliere a caso un numero minore di $p-1$ e continuare sinchè non ci si imbatte in una radice primitiva modulo p. Il seguente test permette appunto di decidere se un intero a è una radice primitiva modulo p, ove p è un primo.

Proposizione 4.13 *Sia $p > 1$ un primo e sia $p - 1 = q_1^{\alpha_1} \cdots q_r^{\alpha_r}$ la fattorizzazione in primi di $p - 1$. Allora un intero a è una radice primitiva modulo p se e solo se $a^{\frac{p-1}{q_i}} \not\equiv 1$ mod p per ogni $i = 1, \ldots, r$.*

Dimostrazione. se a è una radice primitiva modulo p allora non può succedere che $a^{\frac{p-1}{q_i}} \equiv 1 \bmod p$ per qualche i, perchè in tal caso l'ordine di $[a]$ in $(\mathbb{Z}/p\mathbb{Z})^*$ sarebbe minore di $p - 1$.
Viceversa, supponiamo che $a^{\frac{p-1}{q_i}} \not\equiv 1 \bmod p$ per ogni $i = 1, \ldots, r$. Sia s l'ordine di $[a]$ in $(\mathbb{Z}/p\mathbb{Z})^*$. Per il teorema di Lagrange 3.7 si ha che $s|p - 1$, quindi $s = q_1^{\beta_1} \cdots q_r^{\beta_r}$, con $0 \leq \beta_i \leq \alpha_i$ per ogni $i = 1, \ldots, r$. Se per qualche i fosse

$\beta_i < \alpha_i$ allora si avrebbe che $q_i s | p-1$, quindi $a^{\frac{p-1}{q_i}} = (a^s)^{\frac{p-1}{q_i s}} \equiv 1 \bmod p$, assurdo. □

Vediamo ora l'implementazione di un tale test in Maple. Si noti che è necessario conoscere la fattorizzazione di $p-1$, il che non è banale, e ci occuperemo in seguito di questo problema. In ogni caso, per interi "piccoli" abbiamo a disposizione il comando ifactors, per cui applichiamo il nostro test nei casi in cui tale comando ci abbia fornito una fattorizzazione di $p-1$.

```
rapri:=proc(a,p)
local L,n,i,e;
L:=op(2,ifactors(p-1));
n:=true;
for i from 1 to nops(L) while n do
  e:= (p-1)/op(1,op(i,L));
  if pow(a,e,p)=1 then n:=false
  fi;
od;
n;
end;
```

4.3 Numeri pseudo-casuali

Utilizzando una radice primitiva a modulo p, ove p è un primo, si possono costruire in modo piuttosto semplice delle funzioni che generano numeri pseudo-casuali. Un generatore di numeri pseudo-casuali è un algoritmo deterministico che a partire da una successione di k numeri casuali, dà come output una successione di $l >> k$ numeri che "sembrano" casuali. Ciò che si richiede in genere è che non ci sia nessun algoritmo polinomiale che sia in grado di stabilire, con probabilità significativamente maggiore di $1/2$, se la successione è stata generata con un generatore di numeri pseudo-casuali o è effettivamente casuale.

Nel nostro caso, indicando con $\langle x \rangle$ il resto della divisione di x per p, si ha che per $n = 1, \ldots p-1$, i numeri $\langle a^n \rangle$ coincidono con i numeri $1, 2, \ldots, p-1$ in un altro ordine, sostanzialmente imprevedibile. Partendo da questa osservazione, si possono costruire generatori di numeri pseudo-casuali.

Inoltre, un modo di generare numeri razionali pseudo-casuali appartenenti all'intervallo $[0, 1)$ può essere quello di considerare l'applicazione

$$f(n) = \frac{\langle a^n \rangle - 1}{p-1},$$

purchè si scelga p abbastanza grande (la successione $f(n)$ è infatti periodica di periodo $p-1$).

Per una trattazione più dettagliata sui numeri pseudo-casuali si veda [10, Capitolo 5].

Capitolo 5

Pseudoprimi e test di primalità

5.1 Pseudoprimi e numeri di Carmichael

Definizione 5.1 *Sia $b > 1$ un intero. Un intero positivo n tale che* $\mathrm{MCD}(n, b) = 1$ *si dice uno* pseudoprimo per la base b *se n è composto e*

$$b^{n-1} \equiv 1 \mod n. \tag{5.1}$$

La condizione 5.1 è detta *condizione di Fermat.*

Osservazione. Si noti che se n non soddisfa la condizione di Fermat per qualche base b, allora n non è primo, per il terorema di Fermat 3.6, esistono però interi composti per cui vale la condizione di Fermat in qualche base. Ad esempio $341 = 11 \cdot 31$ soddisfa $2^{340} \equiv 1 \mod 341$, $91 = 7 \cdot 13$ soddisfa $3^{90} \equiv 1 \mod 91$, e si ha anche che $100^{560} \equiv 1 \mod 561$.

Dati due interi n e b maggiori di 1, diremo che n passa il *test di pseudoprimalità* per la base b se verifica 5.1. Quindi questo test è idoneo a stabilire se un dato intero è composto, ma non a certificare che è primo.

Proposizione 5.1 *Per ogni intero $b > 1$, esistono infiniti pseudoprimi in base b.*

Dimostrazione. Sia p un numero primo dispari tale che $p \nmid b$ e $p \nmid (b^2 - 1)$. Chiaramente, ci sono infiniti primi con questa proprietà. Sia

$$n = \frac{b^{2p} - 1}{b^2 - 1} = b^{2p-2} + b^{2p-4} + \ldots + b^2 + 1.$$

Dimostriamo che n è uno pseudoprimo per la base b.
Si ha che $n - 1$ è somma di un numero pari di termini (perchè $p - 1$ è pari) tutti pari o tutti dispari (a seconda che b sia pari o dispari), quindi $n - 1$ è

pari. Inoltre, dalla definizione di n di ottiene $(n-1)(b^2-1) = b^2(b^{2p-2}-1)$. Ora $b^{2p-2} = (b^{p-1})^2 \equiv 1 \bmod p$ per il teorema di Fermat, e $p \nmid (b^2-1)$ per ipotesi, quindi $p|(n-1)$. Poichè p è dispari e $n-1$ è pari, si ha $n-1 = 2pk$ per qualche intero k. Poichè $n(b^2-1) = b^{2p}-1$ segue che $b^{2p} \equiv 1 \bmod n$, quindi $b^{n-1} = (b^{2p})^k \equiv 1 \bmod n$, come richiesto. Poichè $n = b^{2p-2}+b^{2p-4}+\ldots+b^2+1$ si ha che $\mathrm{MCD}(n,b) = 1$. Inoltre $n = \frac{a^p-1}{a-1} \times \frac{a^p+1}{a+1}$ è composto, in quanto entrambi i fattori sono interi e maggiori di 1, essendo p dispari. □

Proposizione 5.2 *Si n un intero composto. Se n non è uno pseudoprimo per almeno una base b, allora non è uno pseudoprimo per almeno $\varphi(n)/2$ basi.*

Dimostrazione. Si noti che le basi interessanti sono $\varphi(n)-1$, cioè sono gli interi b con $1 < b < n$ e $\mathrm{MCD}(n,b) = 1$.
Sia $\mathcal{B} = \{b \in (\mathbb{Z}/n\mathbb{Z})^* : b^{n-1} = [1]\}$ l'insieme delle basi per cui n è uno pseudoprimo (viste modulo n) a cui abbiamo aggiunto l'elemento [1]. 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 tali che $a^{n-1} = b^{n-1} = [1]$, segue che $(ab)^{n-1} = [1]$ e $(a^{-1})^{n-1} = (a^{n-1})^{-1} = [1]$, quindi $ab, a^{-1} \in \mathcal{B}$. Inoltre per ipotesi esiste almeno una ase b rispetto a cui n non è pseudoprimo, quindi $b \notin \mathcal{B}$. Poichè l'ordine di $\mathcal{B}$ è un divisore dell'ordine di $(\mathbb{Z}/n\mathbb{Z})^*$ per il teorema di Lagrange 3.7 si ha che $\frac{|(\mathbb{Z}/n\mathbb{Z})^*|}{|\mathcal{B}|} \geq 2$, quindi $|\mathcal{B}| \leq \varphi(n)/2$. □

Definizione 5.2 *Un intero positivo n si dice* numero di Carmichael *se è uno pseudoprimo rispetto a qualunque base ammissibile $b > 1$.*

Osservazione. Si noti che dire che n è un numero di Carmichael equivale a dire che n non è primo e $[b]_n^{n-1} = [1]_n$ per ogni $[b]_n \in (\mathbb{Z}/n\mathbb{Z})^*$.

Proposizione 5.3 *Sia n un intero composto dispari. Allora n è un numero di Carmichael se e solo se $n = p_1 p_2 \cdots p_k$, ove i p_i sono primi distinti, $k \geq 2$ e $p_i - 1$ divide $n-1$ per ogni i.*

Dimostrazione. Supponiamo che n sia un numero di Carmichael dispari e sia $n = p^\alpha \cdot m$, $\alpha \geq 1$. Per le proposizioni 3.1 e 3.2 si ha che $(\mathbb{Z}/n\mathbb{Z})^*$ è isomorfo a $(\mathbb{Z}/p^\alpha\mathbb{Z})^* \times (\mathbb{Z}/m\mathbb{Z})^*$. Per il teorema 4.12 esiste una radice primitiva b modulo p^α. Si noti che l'ordine di $[b]_{p^\alpha}$ divide l'ordine di $[b]_n$. Poichè n è di Carmichael, si ha che $[b]_n^{n-1} = [1]_n$, quindi l'ordine di $[b]_n$ divide $n-1$. Dunque $p^{\alpha-1}(p-1)$ divide $n-1$. Segue che $p-1$ divide $n-1$. Segue anche che $p^{\alpha-1}|n-1$, ma poichè $p^{\alpha-1}|n$ si ha che $p^{\alpha-1}|n-(n-1) = 1$, quindi $\alpha = 1$. Quindi nella scomposizione in primi di n ogni primo p_i compare con esponente 1, $p_i - 1|n-1$ e compaiono almeno 2 primi, in quanto n è composto.
Supponiamo ora che n sia della forma $n = p_1 p_2 \cdots p_k$, ove i p_i sono primi distinti, $k \geq 2$ e $p_i - 1$ divide $n-1$ per ogni i. Per le proposizioni 3.1 e 3.2 si ha che $(\mathbb{Z}/n\mathbb{Z})^*$ è isomorfo a $(\mathbb{Z}/p_1\mathbb{Z})^* \times \cdots \times (\mathbb{Z}/p_k\mathbb{Z})^*$. È quindi sufficiente dimostrare che $([b_1]_{p_1}, \ldots, [b_k]_{p_k})^{n-1} = ([1]_{p_1}, \ldots, [1]_{p_k})$ per ogni elemento $([b_1]_{p_1}, \ldots, [b_k]_{p_k}) \in (\mathbb{Z}/p_1\mathbb{Z})^* \times \cdots \times (\mathbb{Z}/p_k\mathbb{Z})^*$. Poichè $p_i - 1$ divide $n-1$

per ogni i, si ha che $n-1=(p_i-1)s_i$, quindi per il teorema di Fermat 3.6 $([b_i]_{p_i})^{n-1}=(([b_i]_{p_i})^{p_i-1})^{s_i}=[1]_{p_i}$ per ogni i, come volevasi dimostrare. □

Osservazione. Si noti che se n è pari, allora non è un numero di Carmichael. Infatti, preso $a=n-1$, abbiamo che $[n-1]_n^{n-1}=[-1]_n^{n-1}=[1]_n$ se e solo se $n-1$ è pari, cioè se e solo se n è dispari. Quindi $(n-1)^{n-1}\equiv 1 \bmod n$ se e solo se n è dispari.

Proposizione 5.4 *Un numero di Carmichael è prodotto di almeno* 3 *primi distinti.*

Dimostrazione. Supponiamo per assurdo che un numero di Carmichael m sia prodotto di 2 primi $n=qp$, con $p<q$ primi distinti. Dalla proposizione precedente sappiamo che $p-1$ e $q-1$ dividono $n-1$. Ora $n-1=(p-1)q+(p-1)+(q-1)$, quindi $p-1|n-1$ implica $p-1|q-1$ e $q-1|n-1$ implica $q-1|p-1$. Segue che $p-1=q-1$, cioè $p=q$, una contraddizione. □

Esempi. Si ha che $561=3\cdot 11\cdot 17$ è un numero di Carmichael, e così pure $29341, 172081$ e 564651361.

Osservazione. E' stato dimostrato da Alford, Granville e Pomerance che esistono infiniti numeri di Carmichael. Inolte, se x è sufficientemente grande, si ha che

$$C(x)=|\{n\leq x\colon n \text{ è di Carmichael}\}|\geq x^{\frac{2}{7}}.$$

Definizione 5.3 *Sia $n>1$ un intero dispari, $n=p_1^{\alpha_1}\cdots p_r^{\alpha_r}$, ove i p_i sono primi distinti. Si dice* funzione di Carmichael di n, *e si indica con $\lambda(n)$, il minimo comune multiplo dei $\varphi(p_i)$, per $i=1,\dots,r$.*

Proposizione 5.5 *Sia $n>1$ un intero dispari. Allora $\lambda(n)|\varphi(n)$ e $\lambda(n)$ è l'esponente di $(\mathbb{Z}/n\mathbb{Z})^*$.*

Dimostrazione. Per le proposizioni 3.1 e 3.2 si ha che $(\mathbb{Z}/n\mathbb{Z})^*$ è isomorfo a $(\mathbb{Z}/p_1^{\alpha_1}\mathbb{Z})^*\times\cdots\times(\mathbb{Z}/p_k^{\alpha_k}\mathbb{Z})^*$. Per il teorema 4.9 si ha che il gruppo $(\mathbb{Z}/p_1^{\alpha_1}\mathbb{Z})^*$ è ciclico per ogni i, e sia $[b_i]_{p_i^{\alpha_i}}$ un suo generatore.
Si ha che l'elemento $([b_1]_{p_1^{\alpha_1}},\dots,[b_k]_{p_k^{\alpha_k}})$ ha ordine esattamente $\lambda(n)$, e per il teorema di Lagrange 3.7 si ha che $\lambda(n)$ divide $|(\mathbb{Z}/n\mathbb{Z})^*|=\varphi(n)$.
Sia $([b_1^{r_1}]_{p_1^{\alpha_1}},\dots,[b_k^{r_k}]_{p_k^{\alpha_k}})$ un generico elemento di $(\mathbb{Z}/p_1^{\alpha_1}\mathbb{Z})^*\times\cdots\times(\mathbb{Z}/p_k^{\alpha_k}\mathbb{Z})^*$, e sia r il suo ordine. Si ha che $([b_1^{r_1}]_{p_1^{\alpha_1}},\dots,[b_k^{r_k}]_{p_k^{\alpha_k}})^{\lambda(n)}=([1]_{p_1^{\alpha_1}},\dots,[1]_{p_k^{\alpha_k}})$, quindi $r|\lambda(n)$. In particolare $r\leq\lambda(n)$, come volevasi. □

Teorema 5.6 (Teorema di Carmichael) *Sia $n>1$ un intero dispari, e b un intero tale che* $\mathrm{MCD}(n,b)=1$. *Allora $b^{\lambda(n)}\equiv 1$ mod n.*

Dimostrazione. Segue immediatamente dalla dimostrazione della proposizione precedente, in quanto l'ordine r di un qualsiasi elemento di $(\mathbb{Z}/n\mathbb{Z})^*$ divide $\lambda(n)$. □

5.2 Alcuni test di primalità

Diamo ora un criterio per stabilire se un intero p è primo, supponendo di avere a disposizione la fattorizzazione di $p-1$. Tale criterio fu enunciato da Lucas nel 1881 in una forma leggermente più debole e dimostrato da Lehmer nel 1927.

Teorema 5.7 *Sia $n \geq 3$ un intero. Se esiste un intero a tale che $a^{n-1} \equiv 1$ mod n ma $a^{\frac{n-1}{q}} \not\equiv 1$ mod n per ogni primo q che divide $n-1$, allora n è primo.*

Dimostrazione. Sia s l'ordine di $[a]_n$ in $(\mathbb{Z}/n\mathbb{Z})^*$. Dalla prima ipotesi segue che $s|n-1$ Sia quindi $n-1 = k \cdot s$ e supponiamo per assurdo che sia $k > 1$. Sia q un primo che divide k, allora $q|(n-1)$. Si ha che $a^{\frac{n-1}{q}} = (a^s)^{k/q} \equiv 1^{k/q} = 1$ mod n, contraddicendo la seconda ipotesi. Quindi $s = n-1$. D'altra parte per il teorema di Lagrange 3.7 si ha che $s|\varphi(n)$. Segue che $n-1 = s \leq \varphi(n) \leq n-1$, quindi $\varphi(n) = n-1$. Ricordando la formula per $\varphi(n)$, si conclude che n deve essere primo. □

Il test precedente richiede che si conoscano tutti i primi che dividono $n-1$. Presentiamo ora un test in cui si richiede un pò di meno che avere a disposizione la fattorizzazione completa di $n-1$.

Teorema 5.8 (Pocklington) *Sia $n \geq 3$ un intero e sia $n-1 = FR$, ove tutti i fattori primi di F sono noti, $F \geq \sqrt{n}$ e $\mathrm{MCD}(F,R) = 1$. Se esiste un intero a tale che $a^{n-1} \equiv 1$ mod n e $\mathrm{MCD}(a^{\frac{n-1}{q}} - 1, n) = 1$ per ogni primo q che divide F, allora n è primo.*

Dimostrazione. Sia q un primo che divide F, e sia q^α la massima potenza di q che divide F. Se p è un primo che divide n, dalle ipotesi segue che $a^{n-1} \equiv 1$ mod p ma $a^{\frac{n-1}{q}} \not\equiv 1$ mod p. Sia s l'ordine di $[a]_p$ in $(\mathbb{Z}/p\mathbb{Z})^*$. Si ha che $s|n-1$ ma $s \nmid \frac{n-1}{q}$, quindi $q^\alpha|s$. Poichè questo è vero per ogni primo q che divide F, si ha che $F|s$. Inoltre $s|(p-1)$ per il teorema di Fermat, quindi $F|(p-1)$. Segue che se un primo p divide n si ha che $p \geq F+1$, dunque per la proposizione 2.1 n non può essere composto. □

Corollario 5.9 (Teorema di Proth) *Sia $n \geq 3$ un intero della forma $n = k \cdot 2^m + 1$, con $m \geq 2$, k è dispari e $k < 2^m$. Se esiste un intero a tale che $a^{\frac{n-1}{2}} \equiv -1$ mod n, allora n è primo.*

Dimostrazione. Segue dal teorema 5.8, ponendo $F = 2^m$ e $R = k$. □

Modificando leggermente il teorema 5.7 si ottiene quello che comunemente si indica con *certificato di primalità succinto*, cioè una breve dimostrazione della primalità di un intero.

Teorema 5.10 *Sia $n \geq 3$ un intero dispari. Se esiste un intero a tale che $a^{\frac{n-1}{2}} \equiv -1$ mod n ma $a^{\frac{n-1}{2q}} \not\equiv -1$ mod n per ogni primo dispari q che divide $n-1$, allora n è primo. Viceversa, se n è primo allora questa condizione è verificata da ogni radice primitiva modulo n.*

Dimostrazione. Supponiamo che $a^{\frac{n-1}{2}} \equiv -1 \bmod n$. Allora si ha anche che $a^{n-1} \equiv 1 \bmod n$, e si può applicare il teorema 5.7. Resta quindi da dimostrare che $a^{\frac{n-1}{q}} \not\equiv 1 \bmod n$ per ogni primo dispari q che divide $n-1$. Sia $m = a^{\frac{n-1}{2q}}$; allora per ipotesi si ha che $m^q \equiv -1 \bmod n$. Se fosse anche $m^2 \equiv 1 \bmod n$ si avrebbe che $m = m^{q-(q-1)} = m^q (m^{-2})^{\frac{q-1}{2}} \equiv -1 \bmod n$, contro l'ipotesi.
Il viceversa è immediato.□

5.3 Pseudoprimi forti e residui quadratici

Il nostro prossimo scopo è di descrivere il test di Miller. Abbiamo visto che, a causa dell'esistenza dei numeri di Carmichael, la condizione di Fermat non è sufficiente per stabilire la primalità di un numero. È necessaria quindi una condizione più forte della pseudoprimalità, che viene suggerita dalla seguente osservazione.

Teorema 5.11 *Sia n un primo dispari, $n-1 = 2^s t$, con t dispari. Se a è un intero tale che* $\mathrm{MCD}(a,n) = 1$ *si ha che*

$$a^t \equiv 1 \mod n,$$

oppure esiste un intero $r \in \{0, 1, \ldots, s-1\}$ tale che

$$a^{2^r t} \equiv -1 \mod n.$$

Dimostrazione. Poichè n è primo, $\mathbb{Z}/n\mathbb{Z}$ è un campo. Utilizzando più volte l'uguaglianza $[x]^2 - [y]^2 = ([x]-[y])([x]+[y])$ e il teorema di Fermat si ha che $[0] = [a]^{n-1} - [1] = [a]^{2^s t} - [1] = ([a]^{2^{s-1}t} + [1])([a]^{2^{s-1}t} - [1]) = ([a]^{2^{s-1}t} + [1]) \cdots ([a]^{2^2 t} + [1])([a]^{2t} + [1])([a]^t + [1])([a]^t - [1])$, quindi almeno uno dei fattori deve essere uguale a $[0]$, come volevasi dimostrare. □

Questo teorema suggerisce la seguente definizione:

Definizione 5.4 *Sia $n > 2$ un numero composto dispari e $b > 1$ un intero tale che* $\mathrm{MCD}(b,n) = 1$. *Sia $n-1 = 2^s t$, con t dispari. Si dice che n è uno* pseudoprimo forte *per la base b se n verifica una delle seguenti condizioni:*

$$b^t \equiv 1 \mod n,$$

oppure esiste un intero $r \in \{0, 1, \ldots, s-1\}$ tale che

$$b^{2^r t} \equiv -1 \mod n.$$

Osservazione. Si vede facilmente che ognuna di queste condizioni esclude tutte le altre, quindi se n è uno pseudoprimo forte per la base b verifica una sola di esse. Inoltre, se n è uno pseudoprimo forte per la base b, allora è a maggior ragione uno pseudoprimo per la base b.

Definizione 5.5 *Diremo che un intero n composto* passa il test di Miller per la base b, *ove* $\mathrm{MCD}(b,n)=1$, *se n è uno pseudoprimo forte per la base b.*

Proposizione 5.12 *Siano n e b interi, con $n>1$ e $\mathrm{MCD}(n,b)=1$. Se n non passa il test di Miller per la base b allora n non è primo.*

Dimostrazione. Segue direttamente dal teorema 5.11. □

Pomerance, Selfridge e Wagstaff hanno dimostrato che gli pseuduprimi forti per la base 2 sono più rari degli pseudoprimi per la base 2, come si evince dalla seguente tabella:

n	# di ps-primi $< n$	# di ps-primi forti $< n$
10^3	3	0
10^6	245	46
10^9	5 597	1 282
25×10^9	21 853	4 842

Il test di Miller diventa più efficace se lo implementiamo per basi diverse. Consideriamo contemporaneamente le basi $2,3$ e 5. Rispetto ad esse, il primo pseudoprimo è 1729, e ci sono 2522 pseudoprimi minori di 25×10^9. Il primo pseudoprimo forte per tali basi è 25 326 001, e ci sono solo 13 pseudoprimi minori di 25×10^9. Se consideriamo contemporaneamente le basi $2,3,5$ e 7, ci sono 1770 pseudoprimi minori di 25×10^9 e c'è un solo pseudoprimo minore di 25×10^9, e precisamente 3215031751.

Queste considerazioni ci forniscono un utile test di primalità per gli interi minori di 25×10^9. Se n è un intero minore di 25×10^9 e diverso da 3215031751 e passa il test di Miller per le basi $2,3,5$ e 7, allora n è primo. (Si noti che comunque in tal caso n è ancora sufficientemente "piccolo" da poter utilizzare il metodo di divisione per provarne la primalità).

Si dimostra che non esiste l'analogo dei numeri di Carmichael per la pseudoprimalità forte, cioè non esistono numeri composti che passino il test di Miller per ogni base ammissibile b. Quidi il test di Miller ci fornisce un criterio deterministico per stabilire la primalità di un intero. Per dimostrare questo teorema ci è utile la seguente importante nozione.

Definizione 5.6 *Siano p un numero primo ed n un intero. Si dice che n è un* residuo quadratico modulo p *se* $\mathrm{MCD}(n,p)=1$ *ed esiste un intero t tale che $n\equiv t^2 \ mod\ p$.*

Equivalentemente, n è un residuo quadratico modulo p se e solo se la classe $[n]_p$ è un quadrato in $\mathbb{Z}/pZ$.

Il problema di trovare un algoritmo efficiente per decidere se n è un residuo quadratico modulo p è stato studiato da alcuni dei più grandi matematici tra la fine del '700 e gli inizi dell'800.

La seguente osservazione è dovuta ad Eulero:

Teorema 5.13 *Sia p un primo dispari e sia b un intero positivo non divisibile per p. Allora*

$$b^{\frac{p-1}{2}} \equiv \pm 1 \mod p, \qquad \text{e}$$

$$b^{\frac{p-1}{2}} \equiv 1 \mod p$$

se e solo se b è un residuo quadratico modulo p.

Dimostrazione. La prima affermazione segue dal fatto che per il teorema di Fermat 3.6 si ha che $([b]^{\frac{p-1}{2}})^2 = [1]$ in $\mathbb{Z}/p\mathbb{Z}$, e poichè $\mathbb{Z}/p\mathbb{Z}$ è un campo le sole radici del polinomio $x^2 - 1$ sono $[\pm 1]$.
Supponiamo che esista t tale che $b \equiv t^2 \mod p$. Allora t non può essere divisibile per p e per il teorema di Fermat 3.6 si ha che $b^{\frac{p-1}{2}} \equiv t^{p-1} \equiv 1 \mod p$.
Viceversa,dimostriamo che se b non è un residuo quadratico modulo p allora $b^{\frac{p-1}{2}} \equiv -1 \mod p$. In $(\mathbb{Z}/p\mathbb{Z})^*$ associamo ad ogni classe $[i]$ l'unica classe j tale che $[i][j] = [b]$ (si ha che $[j] = [i]^{-1}[b]$). Poichè b non è un residuo quadratico modulo p, si avrà sempre $[i] \neq [j]$. Scriviamo ora il prodotto di tutti gli elementi di $(\mathbb{Z}/p\mathbb{Z})^*$, accoppiandoli a due a due in modo che il prodotto di ogni coppia sia $[b]$. Si ottiene che $[(p-1)!] = [b]^{\frac{p-1}{2}}$, ma per il teorema di Wilson 3.9 si ha che $[(p-1)!] = [-1]$, da cui la tesi. □

5.4 Il test deterministico di Miller

Abbiamo descritto nella precedente sezione il test di Miller. Vediamo ora finalmente che esso dimostra la primalità di un intero.

Proposizione 5.14 *Se $n > 1$ è un intero composto, allora esiste una base b tale che n non passi il test di Miller per tale base.*

Ci serve prima un lemma:

Lemma 5.15 *Siano r ed s interi positivi e sia $d = \mathrm{MCD}(r, s)$. Sia p un primo e b un intero coprimo con p. Se $b^r \equiv 1$ mod p e $b^s \equiv 1$ mod p allora $b^d \equiv 1$ mod p.*

Dimostrazione. Nell'anello $\mathbb{Z}/p\mathbb{Z}$ si ha che $[b]^r = [1] = [b]^s$. Sia $d = ur + vs$, con $u, v \in \mathbb{Z}$. Si ha che $[b]^d = [b]^{ur+vs} = [1]$, da cui la tesi. □

Dimostrazione della proposizione 5.14. Se $n = p^\alpha$, con $\alpha \geq 2$, o se n è pari, per la proposizione 5.3 e l'ossevazione che la segue, esiste una base b rispetto alla quale n non è uno pseudoprimo, di conseguenza n non è neppure uno pseudoprimo forte rispetto a tale base. *Un modo per bypassare la proposizione 5.3 è il seguente: se $n = p^\alpha$, con $\alpha \geq 2$ e p dispari, sia b una radice primitiva modulo α allora l'ordine di $[b]_n$ è $\varphi(p^\alpha) = p^{\alpha-1}(p-1)$ quindi $[b]_n^{n-1} = [b]_n^{p^\alpha - 1} \neq [1]_n$ in quanto $p | p^{\alpha-1}(p-1) \nmid p^\alpha - 1$.*
Possiamo quindi supporre che n abbia almeno due fattori primi dispari distinti p e q. Siano

$$p = 2^a u + 1,$$

$$q = 2^b v + 1,$$

ove $a, b \geq 1$ e u, v dispari. Possiamo inoltre supporre che $a \leq b$. Sia m un intero tale che $\text{MCD}(m, n) = 1$, m sia un residuo quadratico modulo p ma m non sia un residuo quadratico modulo q (tale m esiste per il Teorema Cinese dei Resti). Dimostriamo che allora n non è uno pseudoprimo forte per la base m. Sia

$$n = 2^s t + 1,$$

ove t è dispari e $s \geq 1$. Supponiamo per assurdo che n sia uno pseudoprimo forte per la base m. Allora in particolare n è uno pseudoprimo per tale base, quindi ogni primo che divide n divide esattamente uno dei fattori

$$m^t - 1, m^t + 1, m^{2t} + 1, \ldots, m^{2^{s-1}t} + 1,$$

e poichè n è uno pseudoprimo forte per la bas m allora tutti i primi che dividono n devono dividere lo stesso fattore.
Notiamo che se un primo dispari divide $m^{2^{j-1}t}+1$ allora divide anche $m^{2^j t}-1 = (m^{2^{j-1}t}+1)(m^{2^{j-1}t}-1)$ ma non divide $m^{2^{j-1}t} - 1 = (m^{2^{j-1}t} + 1) - 2$.
Siano j e k i più piccoli interi positivi tali che

$$p | m^{2^j t} - 1, \quad \text{e} \quad q | m^{2^k t} - 1.$$

Si ha che $0 \leq j, k \leq s$. Poichè m è un residuo quadratico modulo p per il teorema 5.13 si ha che $p | m^{\frac{p-1}{2}} - 1 = m^{2^{a-1}u} - 1$.
Per il lemma 5.15 si ha che $m^{\text{MCD}(2^j t, 2^{a-1}u)} \equiv 1 \bmod p$.
Se fosse $a \leq j$ si avrebbe che $\text{MCD}(2^j t, 2^{a-1}u) = 2^{a-1}\text{MCD}(t, u)$, e quindi $2^{a-1}t$ sarebbe un multiplo di $\text{MCD}(2^j t, 2^{a-1}u)$. Quindi si avrebbe che $m^{2^{a-1}t} \equiv 1 \bmod p$, contraddicendo la minimalità di j.
Quindi $a > j$. Ci occupiamo ora di q. Si ha che q divide $m^{q-1} - 1 = m^{2^b v} - 1$, quindi per il lemma 5.15 q divide $m^{\text{MCD}(2^b v, 2^k t)} - 1$.
Se fosse $b > k$ allora $\text{MCD}(2^b v, 2^k t) = 2^k \text{MCD}(v, t)$, dunque $m^{2^{b-1}v} \equiv 1 \bmod q$. Ma poichè m non è un residuo quadratico modulo q, per il teorema 5.13 $q \nmid m^{\frac{q-1}{2}} - 1 = m^{2^{b-1}v} - 1$, contraddizione.
Quindi $b \leq k$. Riassumendo, si ottiene che $j < a \leq b \leq k$, e quindi j e k non possono essere uguali. Quindi p e q dividono fattori distinti e n non può passare il test di Miller per la base m, come volevasi dimostrare. □

5.5 Test probabilistico di Miller-Rabin

Supponiamo di voler dimostrare che un intero n è primo utilizzando il test deterministico di Miller. Testare n per tutte le basi ammissibili b richiede troppi calcoli, e sarebbe utile sapere che esiste un numero relativamente "piccolo" B (che dipende da n) tale che se n è composto allora n non è uno pseudoprimo forte per almeno una base $b < B$. Questo ci permetterebbe di eseguire il test solo sulle prime B basi ammissibili. Questo è vero se si suppone che l'Ipotesi di

Riemann generalizzata sia vera. Tale famosa congettura è però tuttora aperta (si veda [5] per una sua descrizione più precisa).

Il teorema dimostrato da Miller è il seguente:

Teorema 5.16 *Se vale l'Ipotesi di Riemann generalizzata ed n è un intero composto dispari, allora n non passa il test di Miller per almeno una base b tale che $b < 2\log^2 n$.*

Il seguente teorema dimostra comunque che il test di Miller può essere utilizzato per un test probabilistico di primalità molto efficace. Tale test è detto il test di Miller-Rabin.

Teorema 5.17 (Rabin) *Se $n \geq 3$ è un intero dispari composto, allora l'insieme $\{1, \ldots, n-1\}$ contiene al più $(n-1)/4$ interi b tali che n passi il test di Miller per la base b.*

Per applicare il test di Miller-Rabin ad un intero dispari $n \geq 3$ si procede nel modo seguente. Sia $n - 1 = 2^s t$, con t dispari. Scegliamo a caso un numero $a \in \{2, 3, \ldots, n-1\}$. Se $\mathrm{MCD}(a, n) > 1$ allora n è composto. Altrimenti, applichiamo ad n il test di Miller per la base a. Se n non lo passa, allora è composto. Altrimenti per il teorema di Rabin 5.17 sappiamo che la probabilità che n sia composto è al più $1/4$. Allora scegliamo a caso un altro numero $a \in \{2, 3, \ldots, n-1\}$ e ripetiamo il test. Se ripetiamo il test m volte e se n è n lo passa sempre, allora la probabilità che n sia composto è al più $1/4^m$. Si noti che per $m = 10$ tale probabilità è al più all'incirca $1/10^6$, cioè molto bassa.

Un'analisi più dettagliata del test di Miller-Rabin ha dimostrato che in realtà tale probabilità è ancora inferiore.

Per completezza, concludiamo ora con la dimostrazione del teorema 5.17. Essa richiede la conoscenza di alcune proprietà elementari dei gruppi che non tutti i lettori potrebbero conoscere. Si osservi inoltre tale tale dimostrazione implica in particolare il teorema 5.14.

Ricordiamo primo un risultato di teoria dei gruppi.

Lemma 5.18 *Sia $(G, \cdot, e)$ un gruppo abeliano di esponente p, ove p è un numero primo. Allora l'ordine di G è una potenza di p.*

Dimostrazione. Supponiamo per assurdo che esista un primo q diverso da p che divide l'ordine di G. Allora per il teorema di Cauchy [7, p.42] esiste un elemento g di G di ordine q. D'altra parte, poichè G ha esponente p si ha anche che $g^p = e$, assurdo perchè q non divide p. □

Dimostrazione del Teorema 5.17. Sia $n \geq 3$ un numero composto dispari e sia $n - 1 = 2^s t$, con t dispari. Vogliamo stimare il numero di elementi $a \in \{1, \ldots, n-1\}$ tali che $\mathrm{MCD}(a, n) = 1$ e

$$a^t \equiv 1 \mod n, \tag{5.2}$$

oppure

$$a^{2^r t} \equiv -1 \mod n. \tag{5.3}$$

per qualche intero $r \in \{0, 1, \ldots, s-1\}$.
Se un tale a non esiste, abbiamo terminato. Se invece esiste un tale a, ne esiste uno per cui valga la condizione 5.3. Infatti se a soddisfa la condizione 5.2 allora $-a$ soddisfa la condizione 5.3 ed è sufficiente considerare un opportuno rappresentante della classe $[-a]_n$. Sia k il più grande valore di $r \in \{0, \ldots, s-1\}$ tale che esista un intero a che soddisfa la condizione 5.3 e $\mathrm{MCD}(a, n) = 1$. Sia

$$m = 2^k t.$$

Sia inoltre $n = p_1^{\alpha_1} \cdots p_l^{\alpha_l}$ la fattorizzazione in primi di n. Definiamo i seguenti sottogruppi di $(\mathbb{Z}/n\mathbb{Z})^*$:

$$J = \{[b]_n \in (\mathbb{Z}/n\mathbb{Z})^* : [b]_n^{n-1} = [1]_n\},$$

$$K = \{[b]_n \in (\mathbb{Z}/n\mathbb{Z})^* : [b]_{p_i^{\alpha_i}}^m = [\pm 1]_{p_i^{\alpha_i}} \text{ per ogni } i\},$$

$$L = \{[b]_n \in (\mathbb{Z}/n\mathbb{Z})^* : [b]_n^m = [\pm 1]_n\},$$

$$M = \{[b]_n \in (\mathbb{Z}/n\mathbb{Z})^* : [b]_n^m = [1]_n\}.$$

Si ha

$$M \subseteq L \subseteq K \subseteq J \subseteq (\mathbb{Z}/n\mathbb{Z})^*.$$

Se a è un intero tale che n passi il test di Miller per la base a, allora $[a]_n \in L$. Dimostreremo il teorema mostrando che l'indice di L in $(\mathbb{Z}/n\mathbb{Z})^*$ è almeno 4. (Ricordiamo che l'indice di L in $(\mathbb{Z}/n\mathbb{Z})^*$ è $|(\mathbb{Z}/n\mathbb{Z})^*|/|L| = |(\mathbb{Z}/n\mathbb{Z})^*/L|$).
Poichè il quadrato di ogni elemento di K appartiene ad M, il gruppo quoziente K/M ha esponente 2, quindi per il lemma 5.18 ha ordine una ptenza di 2. Ora K/L è isomorfo ad un quoziente di K/M, quindi ha anch'esso ordine una potenza di 2. Segue che l'indice di L in K è del tipo 2^j, con $j \geq 0$.
Dimostriamo che se n è divisibile per almeno 3 primi distinti p, q ed r allora $j \geq 2$, e quindi in questo caso la tesi del teorema è vera. Siamo rispettivamente p^α, q^β e r^γ le massime potenze di p, q ed r che dividono n.
Nelle nostre ipotesi, esiste un intero a tale che $a^m \equiv -1 \bmod n$.
Per il teorema cinese dei resti 1.9 esistono due interi u e v tali che

$$u \equiv a \mod p^\alpha, \qquad u \equiv 1 \mod q^\beta, \qquad u \equiv 1 \mod r^\gamma, \quad \text{e}$$
$$v \equiv a \mod p^\alpha, \qquad v \equiv a \mod q^\beta, \qquad v \equiv 1 \mod r^\gamma.$$

Si ha che $[a]_n, [au]_n, [av]_n$ sono tre elementi di K che hanno immagini distinte in K/L, quindi $|K/L| \geq 4$.
Supponiamo ora che esistano solo 2 primi distinti p e q che dividono n. Allora per il teorema 5.4 n non è di Carmichael, quindi J è un sottogruppo proprio di $(\mathbb{Z}/n\mathbb{Z})^*$ e dunque l'indice di J in $(\mathbb{Z}/n\mathbb{Z})^*$ è almeno 2. Inoltre, di nuovo per il teorema cinese dei resti 1.9 esiste un intero u tale che

$$u \equiv a \mod p, \qquad u \equiv 1 \mod q.$$

Si ha che $[a]_n$ e $[au]_n$ sono due elementi di K che hanno immagini distinte in K/L, quindi $|K/L| \geq 2$. Quindi l'indice di L in $(\mathbb{Z}/n\mathbb{Z})^*$ è almeno 4.

Resta il caso in cui $n = p^e$ è potenza di un solo primo. Mostriamo che in tal caso J ha ordine $p-1$. Infatti sia $[a]_n \in J$ e sia s il suo ordine. Si ha che s divide $n-1 = p^e - 1 = (p-1)(p^{e-1} + p^{e-2} + \cdots + p + 1)$, e inoltre s divide $\phi(n) = p^{e-1}(p-1)$. Quindi s divide $\mathrm{MCD}(n-1, \phi(n)) = p-1$. Sappiamo per il teorema 4.9 che il gruppo $(\mathbb{Z}/p^e\mathbb{Z})^*$ è ciclico, e sia $[t]_{p^e}$ un suo generatore. Allora $[t]_{p^e}^{p^{e-1}} \in J$ ha ordine $p-1$, quindi J è l' unico sottogruppo di $(\mathbb{Z}/p^e\mathbb{Z})^*$ di ordine $p-1$. Segue che, se $n \neq 9$, l'indice di J in $(\mathbb{Z}/n\mathbb{Z})^*$ è almeno 4. Se $n = 9$ il teorema può essere verificato direttamente. $\square$

5.6 Primi casuali

In molti sistemi crittografici a chiave pubblica è necessario avere a disposizione numeri primi casuali aventi un fissato numero k di cifre binarie. Descriviamo ora la costruzione di tali primi casuali.

Supponiamo di avere un generatore casuale di cifre binarie. Nella pratica, si utilizzano generatori di cifre binarie casuali basati sull'hardware, che fanno uso, ad esempio, della casualità del decadimento radioattivo, o del tempo che intercorre tra quando viene battuto un tasto della tastiera e quando viene battuto quello successivo. (Solitamente, si sottopongono tali generatori a test statistici per stabilire se i fenomeni presi come riferimento siano effettivamente casuali).

Vogliamo ora generare un numero primo dispari di k cifre binarie. A questo scopo, poniano uguali a 1 la prima e l'ultima cifra, e scegliamo a caso le altre $k-1$ cifre binarie, con il generatore di cui sopra. Otteniamo così un numero n, e vogliamo testare la sua primalità. Prima si controlla se n è divisibile per qualche numero primo minore di un numero prefissato B (tipicamente $B = 10^6$). Se non si trova nessun divisore primo di n, si applica il test di Miller-Rabin t volte. La scelta $t = 3$ è già sufficiente a rendere la probabilità di errore minore di $(1/2)^{80}$, se $k \geq 1000$. Se n passa il test, allora lo si considera primo. Si può anche scegliere un B più grande, se l'esecuzione delle divisioni è più efficiente del test di Miller-Rabin.

Se invece n non passa il test (e quindi non è primo), si procede con $n+2$, e così via (vedere anche il paragrafo 1.2).

5.7 Primes is in P

Anche se gli algoritmi probabilistici sono quelli più veloci e tuttora i più utilizzati per testare la primalità di un intero, è doveroso citare il recente risultato di Agrawal, Kayal e Saxena, i quali hanno ideato un algoritmo deterministico che decide se un intero è primo o composto in tempo polinomiale.

Per una sua descrizione, si veda [6, p.119].

Capitolo 6

Fattorizzazione

In questo capitolo descriveremo alcuni algoritmi di fattorizzazione.

Per fattorizzare un intero n, come prima cosa ci si assicura che n non sia primo, applicando uno dei testi di primalità che abbiamo descritto. Inoltre, spesso si verifica che n non abbia fattori piccoli e che non sia una quadrato perfetto. Poi si procede utilizzando un algoritmo di fattorizzazione. Più che trovare direttamente i fattori primi di n, può essere intanto utile disporre di un algoritmo che fattorizzi n nella forma $n = ab$, dove a e b sono fattori propri, e poi procedere a fattorizzare separatamente a e b.

6.1 Divisione per tentativi

Per trovare fattori piccoli di un numero n, si utilizza una lista precalcolata contenente tutti i numeri primi minori o uguali ad un certo numero prefissato L (tipicamente $L = 10^6$). Si prova a dividere n per ciascuno di questi numeri primi, e per ciascun primo p della lista che divide n si calcola $e(p)$, che è il massimo esponente tale che $p^{e(p)}$ divide n. Si scrive poi n nella forma $n = p_1^{e(p_1)} \cdots p_k^{e(p_k)} m$ e si usa qualche algoritmo più potente per fattorizzare m.

Se si volesse fattorizzare completamente n con questo metodo bisognerebbe prendere $L = \sqrt{n}$, e in questo caso il numero delle divisioni necessarie non è significativamente più piccolo di $\sqrt{n}$, e la complessità di questo algoritmo è dunque esponenziale.

6.2 Fattorizzazione alla Fermat

Il primo degli algoritmi moderni di fattorizzazione è dovuto a Fermat. Al giorno d'oggi di solito non viene utilizzato, a meno che non si sappia a priori che l'intero n da fattorizzare ha due fattori entrambi abbastanza vicini a $\sqrt{n}$. Ma ha in sé l'idea chiave che sta alla base di due dei più potenti algoritmi per fattorizzare un intero che abbia fattori primi grandi, il Crivello Quadratico e il Metodo delle Frazioni Continue.

L'idea di Fermat è la seguente: sia n è il numero da fattorizzare; se n può essere scritto come la differenza di due quadrati $n = x^2 - y^2$, allora $n = (x - y)(x+y)$ e siamo riusciti a frovare due fattori propri di n. Inoltre, se supponiamo che n sia dispari, allora ogni fattorizzazione di n nel prodotto di due interi si ottiene in questo modo. Infatti, se $n = ab$, con a e b dispari, ponendo $x = \frac{a+b}{2}$ e $y = \frac{a-b}{2}$ si ha quanto voluto.

L'algoritmo di Fermat lavora nella direzione opposta a quella della divisione per tentativi. In quel caso si consideravano numeri piccoli come candidati per i fattori di n, aumentandoli progressivamente sino a $\sqrt{n}$. Ora invece i candidati sono numeri vicino a $\sqrt{n}$, e li si rimpicciolisce progressivamente.

Descriviamo ora l'algoritmo. Dato n, cerchiamo due interi x ed y tali che $x^2 - y^2 = n$. Si inizia prendendo come x il più piccolo intero maggiore o uguale a $\sqrt{n}$ e si pone $y = 0$. Poi si aumenta progressivamente y sinchè $x^2 - y^2 \leq n$. Se $x^2 - y^2 = n$ l'algoritmo termina, altrimenti si aumenta x di 1, si pone $y = 0$ e si reitera il ciclo. Posto $r = x^2 - y^2 - n$, l'algoritmo termina quando $r = 0$.

L'algoritmo presentato qui di seguito ha il vantaggio che i cicli in esso contenuti non coinvolgono prodotti nè divisioni, cosicchè sono estremamente veloci. Si considerano $u = 2x+1$ e $v = 2y+1$ al posto di x e y. Quando x e y variano di una unità, u e v variano di 2 unità. Inoltre r, che è la variabile che ci interessa, aumenta esattamente di u unità quando x^2 passa ad $(x+1)^2$, e diminuisce di v unità quando y^2 passa a $(y+1)^2$. Riportiamo qui di seguito l'implementazione in Maple.

```
fferm := proc(n)
local x, u,v, r;
x:=ceil(sqrt(n));
u:=2*x+1;
r:=x^2-n;
v:=1;
while r <>0 do
  if r>0 then
    r:=r-v;
    v:=v+2;
  else
    r:=r+u;
    u:=u+2;
  fi;
od;
[(u+v-2)/2,(u-v)/2]; end;
```

Lo svantaggio di questo algoritmo è l'enorme numero di cicli di cui necessita. Se lo si utilizza per fattorizzare un numero primo p, alla fine ci darà come output i fattori 1 e p, dimostrando in questo modo la primalità di p. Ma se utilizzato come test di primalità questo algoritmo è estremamente svantaggioso, infatti esso termina dopo $n - \sqrt{n}$ cicli, molti di più di quelli richiesti dalla divisione per tentativi.

6.3 Algoritmo di Lehman

L'algoritmo di Lehman è un compromesso tra i due algoritmi precedenti, ed è più efficiente di ciascuno dei due. Eccone una breve descrizione. Posto $R = n^{\frac{1}{3}}$ applichiamo la divisione per tentativi per tutti i numeri primi minori o uguali ad R. Questo richiede $O(R)$ divisioni. Se nessuna di esse è esatta, allora n è primo oppure n è il prodotto pq di esattamente due primi, che soddisfano $R < p \leq q < n/R = R^2$.

Crandall e Pomerance hanno dimostrato che se n non è primo è possibile trovare x, y e k interi maggiori o uguali a zero tali che:

$$\begin{cases} x^2 - y^2 = 4kn \qquad \text{con } 1 \leq k \leq R \\ \sqrt{4kn} \leq x \leq \sqrt{4kn} + \frac{\sqrt{n/k}}{4R} \\ p = \min\{\text{MCD}(x+y, n), \text{MCD}(x-y, n)\} \end{cases}$$

Per trovare x e y si procede nel modo seguente. Si pone $k = 1$; poi posto $x_0 = \lceil \sqrt{4kn} \rceil$ ed $x_1 = \lfloor \sqrt{4kn} + \frac{\sqrt{n/k}}{4R} \rfloor$, si cerca se per qualche $x \in [x_0, x_1]$ si ha che $x^2 - 4kn$ è un quadrato perfetto y^2. Se questo accade l'algoritmo termina con il calcolo di $\text{MCD}(x+y, n)$, altrimenti si pone $k = k + 1$, si aggiornano x_0 e x_1 e si ripete il ciclo.

Si dimostra che il costo totale dell'algoritmo è $O(R) = O(n^{\frac{1}{3}})$, ed è dunque esponenziale.

6.4 I successivi miglioramenti

M. Kraitchick (1882-1957) si rese conto che era possibile velocizzare l'algoritmo di Fermat cercando anzichè due interi x ed y che verifichino $x^2 - y^2 = n$, due interi "casuali" x ed y tali che $x^2 \equiv y^2 \bmod n$. Il trovare due tali interi non ci garantisce più una fattorizzazione, ma implica che n divide $x^2 - y^2 = (x+y)(x-y)$, ed abbiamo ora una probabilità del 50% che i divisori di n siano distributi tra i divisori di entrambi questi fattori, cosicchè $\text{MCD}(x-y, n)$ sarà un fattore non banale di n.

Il suo metodo per trovare la coppia (x, y) era abbastanza *ad hoc*. Alcuni anni più tardi, nel 1931, Lehmer e Powers hanno trovato un metodo sistematico per trovare tali coppie usando le frazioni continue. Il loro algoritmo però non è stato particolarmente efficiente sino all'avvento di computer molto veloci. Tra la fine degli anni '60 e i primi anni '70 il progresso tecnologico riguardante l'hardware era talmente avanzato che ne valse la pena di riesaminare l'algoritmo di Lehmer-Powers. Uno dei primi a mettere a punto un algoritmo efficiente fu D. Shanks, utilizzando le frazioni continue e l'idea di Kraitchick;il suo algoritmo si chiama Square Forms Factorization (SFF). Nel 1975, J. Brillhart e M. Morrison hanno pubblicato quella che oggi è la forma standard del metodo delle frazioni continue (CFRAC), sviluppando la linea seguita da Lehmer e Power più che quella di Shanks.

Per circa un decennio il metodo di Brillhart e Morrison è stato il metodo più efficiente per fattorizzare interi grandi con fattori primi grandi, ed è in uso tuttora oggi. Tuttavia è stato soppiantato dal Crivello Quadratico di Carl Pomerance (QS) e dal suo raffinamento ad opera di P. Montgomery, il Multiple Polynomial Quadratic Sieve (MPQS). Tali algoritmi utilizzano un approccio differente per trovare coppie (x, y) tali che $x^2 \equiv y^2 \bmod n$, utilizzando una considerevole quantità di memoria. Il loro successo è la conseguenza della disponibilità odierna di memorie grandi ed economiche.

Nel 1988, J. Pollard ha infrodotto una ulteriore modifica al Crivello Quadratico, ideando il crivello con i campi di numeri (NFS). Tale algoritmo è considerato oggi quello più efficiente, per n sufficientemente grande.

Accenniamo ora alla complessità computazionale degli algoritmi sopra menzionati. Tipicamente, per gli algoritmi di fattorizzazione essa viene descritta dalla funzione

$$\mathrm{L}_n[u, v] = e^{v(\log n)^u (\log\log n)^{1-u}},$$

ove n, u, v sono numeri reali con n maggiore della costante di Eulero e.

Si ha che

$$\mathrm{L}_n[0, v] = e^{v(\log\log n)} = (\log n)^v,$$
$$\mathrm{L}_n[1, v] = e^{v(\log n)}.$$

Se l'algoritmo ha complessità computazionale $\mathrm{L}_n[0, v]$ allora è polinomiale, quindi efficiente (anche se in realtà l'efficienza dipende dall'esponente del polinomio), se invece ha complessità computazionale $\mathrm{L}_n[1, v]$ allora è esponenziale. Se invece si ha che $0 < u < 1$ allora la complessità è subesponenziale, cioè l'algoritmo è più veloce di uno esponenziale ma più lento di uno polinomiale.

La divisione per tentativi è esponenziale, mentre la complessità computazionale del crivello quadratico non è ancora stata dimostrata in modo rigoroso, ma si stima che sia del tipo $\mathrm{L}_n[1/2, 1 + o(1)]$.

L'algoritmo più efficiente la cui complessità computazionale è stata dimostrata rigorosamente è SFF, la cui complessità è anch'essa $\mathrm{L}_n[1/2, 1 + o(1)]$, anche se in pratica il Crivello Quadratico si è rivelato più veloce.

Un altro metodo di fattorizzazione è quello delle curve ellittiche (ECM), un algoritmo probabilistico che viene utilizzato per trovare fattori notevolmente più piccoli di $\sqrt{n}$. La sua complessità è $\mathrm{L}_p[1/2, \sqrt{1/2}]$, ove p è il fattore primo più piccolo di n. Per trovare fattori della stessa grandezza di $\sqrt{n}$, comunque, il Crivello Quadratico è più efficiente di ECM.

Infine si è dimostrato che, sotto ipotesi opportune, il crivello con i campi di numeri ha complessità computazionale $\mathrm{L}_n[1/3, (64/9)^{1/3}]$, quindi la sua complessità è molto più vicina a quella polinomiale di quanto non lo sia quella del Crivello Quadratico.

6.5 Il metodo Rho di Pollard

Uno degli aspetti negativi sia del metodo delle frazioni continue che del crivello quadratico è che non sono più veloci a trovare fattori di n di modesta grandez-

za, diciamo compresa tra 10^5 e 10^{10}, di quanto non lo siano a trovare fattori veramente grandi.

Se un numero conmposto ha un divisore primo compreso tra 10^5 e 10^{10}, che è troppo grande per essere trovato con il metodo di divisione per tentativi, ed è ancora piccolo perchè valga la pena di utilizzare un algoritmo più potente, si può utilizzare il metodo Rho di Pollard.

Il primo passo è quello di scegliere una applicazione di $\mathbb{Z}/n\mathbb{Z}$ in sè che si calcoli facilmente, ad esempio sia data dal polinomio $f(x) = x^2 + c$ (tipicamente si prende $c = 1$). Poi si sceglie un intero $x = x_0$ (tipicamente $x_0 = 1$ oppure 2 o un intero a caso) e si calcolano successivamente $x_1 = f(x_0)$, $x_2 = f(x_1) = f(f(x_0))$, $x_3 = f(x_2)$, e così via, cioè si pone $x_{j+1} = f(x_j)$, per $j = 0, 1, 2, \ldots$.

Sia d un divisore non banale di n (ovviamente d non è noto). Poichè le classi resto modulo d sono in numero finito, ad un certo punto si troveranno $x_{\bar{r}}$ e $x_{\bar{s}}$ con $\bar{r} \neq \bar{s}$ tali che $x_{\bar{r}} \equiv x_{\bar{s}} \bmod d$. Da questo punto in poi si avrà chiaramente

$$x_{\bar{r}+t} \equiv x_{\bar{s}+t} \qquad \text{per ogni } t \geq 0, \tag{6.1}$$

cosicchè la nostra successione diventa un ciclo, e la sua rappresentazione grafica è simile alla lettera greca ρ, da cui il nome del metodo (in questo caso la coda della lettera ρ finisce in corrispondenza di $x_{\bar{s}}$, se $\bar{r} > \bar{s}$).

Ciò che ci interessa, comunque, è che c'è un'ottima probabilità che $x_{\bar{r}}$ e $x_{\bar{s}}$ non siano congrui modulo n, e quindi $\mathrm{MCD}(n, x_{\bar{r}} - x_{\bar{s}})$, che è divisibile per d, è un fattore proprio di n. Se la lunghezza del ciclo è c, una volta che siamo fuori dalla coda, qualsiasi coppia (x_j, x_i) tale che $c|i - j$ va bene.

Una prima idea è quella di calcolare sistematicamente tutte le possibili differenze del tipo $x_i - x_j$, per $i > j$, e i corrispondenti $\mathrm{MCD}(n, x_i - x_j)$, sino a che non si trova il divisore proprio cercato. Si calcolano cioè $x_1 - x_0$, $x_2 - x_1$, $x_2 - x_0$, $x_3 - x_2$, $x_3 - x_1$, $x_3 - x_0$, e così via.

Se p è il più piccolo fattore primo di n, il numero di iterazioni atteso prima che si trovi una ripetizione modulo p può essere studiato con la teoria della probabilità (si vesa ad esempio [4, proposizione V.2.1]), e se f si comporta come una funzione "media" di $\mathbb{Z}/n\mathbb{Z}$ in sè è dell'ordine di $p^{1/2} \leq n^{1/4}$. Poichè va calcolato un massimo comun divisore per ogni coppia (i, j) il numero di iterazioni è $O(n^{1/2})$, e quindi piuttosto elevato.

È però possibile migliorare l'algoritmo nel modo seguente. Dato k, sia $h+1$ il suo numero di cifre binarie, cioè $2^h \leq k < 2^{h+1}$. Si calcola dunque $x_k - x_{2^h-1}$ e se $\mathrm{MCD}(n, x_k - x_{2^h-1}) = 1$ abbiamo finito, altrimenti si passa a $k+1$ e così via.

In queso modo per ogni k si calcola una sola differenza del tipo $x_k - x_j$ che abbia k come primo termine, e la complessità computazionale è $O(n^{1/4})$. Resta da dimostrare che se esiste una coppia del tipo $(\bar{s}, \bar{r})$ tale che $\mathrm{MCD}(n, x_{\bar{r}} - x_{\bar{s}}) \neq 1$ allora ne esiste anche una del tipo $k, 2^h - 1$, con $\mathrm{MCD}(n, x_k - x_{2^h-1}) \neq 1$. Se $\bar{r}$ ha h cifre binarie, sia $j = 2^h - 1$ e sia $k = 2^h - 1 + (\bar{r} - \bar{s})$. Si ha così che j è il più grande intero con h cifre binarie e k è un intero con $h+1$ cifre binarie tale che $\mathrm{MCD}(n, x_k - x_{2^h-1}) = \mathrm{MCD}(x_{2^s-1+(\bar{r}-\bar{s})} - x_{2^h-1}) \neq 1$. Si noti che $k < 2^{h+1} = 4 \cdot 2^{h-1} \leq 4\bar{r}$.

Questo algoritmo può fallire per due motivi. Uno è che il primo massimo comun divisore diverso da 1 trovato sia proprio n. In tal caso si cambia la scelta del polinomio $f(x)$ o la scelta di x_0 (tipicamente si cambia c) e si riparte con l'algoritmo. L'altro motivo è che potrebbe essere necessario un tempo molto lungo per trovare un divisore (anche perchè a priori non si conosce la grandezza del più piccolo divisore primo di n). Dopo un tempo sufficientemente lungo, può essere conveniente interrompere l'algoritmo e provare con un diverso polinomio $f(x)$ o cambiare addiruttura algoritmo.

6.6 Il metodo $p-1$ di Pollard

Questo metodo è molto simile a quello precedente. Supponiamo che il numero n da fattorizzare abbia un fattore primo p con la proprietà che i primi che dividono $p-1$ sono piccoli, diciamo minori di un certo numero prefissato B. Si costruisce un numero k tale che k sia multiplo di tutti o della maggior parte degli interi minori di B (ad esempio $k = B!$ oppure k è il minimo comune multiplo di tutti gli interi minori o uguali a B), cosicchè k risulta essere un multiplo di $p-1$. Si sceglie poi un intero a tale che $\mathrm{MCD}(a,n) = 1$ (ad esempio $a = 2$) e si calcola $d = \mathrm{MCD}(a^k - 1, n)$. Se $d \neq 1$ e $d \neq n$ l'algoritmo termina, altrimenti si sceglie un nuovo a oppure un nuovo k (in particolare se $d = 1$ vuol dire che $p \nmid a^k - 1$, quindi $p - 1 \nmid k$ e si cambia k, se $d = n$ allora si ha che $|[a]_p|$ divide k e si prova intanto a cambiare a).

Il principio su cui si basa questo algoritmo è che se abbiamo scelto bene k, e quindi k è effettivamente un multiplo di $p-1$, allora si ha che $a^k \equiv 1 \bmod p$, e quindi p divide $a^k - 1$.

Il lato negativo è che se $p-1$ ha solo fattori grandi, l'algoritmo potrebbe non terminare mai.

Si noti che gli algoritmi di Pollard descritti sopra sono entrambi probabilistici. È troppo dispendioso fattorizzare numeri da 20 a 30 cifre, per non parlare di quelli da 80 a 100, con metodi deterministici. Cioò significa che in certi casi potremmo essere sfortunati e non riuscire a fattorizzare n.

Se applichiamo gli algortmi di Pollard e falliamo, potrebbe essere perchè non ci sono divisori primi negli intervalli appropriati o semplicemente perchè siamo stati sfortunati. Per cambiare la fortuna avversa, si cambiano allora i parametri c, x_0, a, k da cui dipendono gli algoritmi, sperando di essere più fortunati.

Per quanto tempo vale la pena cercare fattori di grandezza media prima di far entrare in gioco algoritmi quali il crivello quadratico? Questa, dice il Bressoud [1], è più un'arte che una scienza, ma vale comuque la pena di spendere qualche minuto variando i paramentri o gli algoritmi, prima di utlizzare algoritmi pù potenti.

Il problema di analizzare la complessità degli algoritmi e di ottimizzare le strategie è molto più difficile con gli algoritmi probabilistici che con quelli deterministici, e molto di quello che servirebbe deve essere ancora dimostrato. Entrano in gioco la ditribuzione dei numeri primi, e, dato un intero n, la distribuzione e la grandezza più probabile dei suoi fattori.

6.7 Il metodo di Dixon e le basi di fattori

Il crivello quadratico, e la maggior parte degli algoritmi subesponenziali sopra menzionati, si basano sull'idea di Kraitchick di considerare congruenze del tipo $x^2 \equiv y^2 \bmod n$, cosicchè si abbia una probabilità ragionevole che $\mathrm{MCD}(x, y)$ sia un fattore proprio di n. La differenza tra i vari algoritmi consiste spesso nella modalità con cui gli interi x^2 e y^2 vengono determinati.

L'idea di Dixon è quella di considerare un intero a caso r e considerare l'intero $g(r)$ tale che $0 \leq g(r) < n$ e $g(r) \equiv r^2 \bmod n$. Se riusciamo a fare in modo che $g(r) = y^2$ sia un quadrato, ecco che si ottiene la congruenza $y^2 \equiv r^2 \bmod n$, che è del tipo cercato.

Per fattorizzare i vari $g(r)$ si sceglie prima una *base di fattori*, cioè un insieme B costituito da primi abbastanza "piccoli", e si procede per tentativi dividendo i vari $g(r)$ per i primi p di B. Ad esempio Dixon aveva preso come base di fattori l'insieme di tutti i numeri primi minori di 10 000, che sono in tutto 1229.

Molti $g(r)$ non si fattorizzeranno completamente, e anche se lo facessero è altamente improbabile che $g(r)$ sia un quadato perfetto, come vorremmo. Si continua a scegliere r a caso un numero molto grande di volte, sino a che non si trovano k numeri r che si fattorizzano completamente, con $k > B$. A questo punto si associa ad ogni r il vettore

$$v(r) = (a_1, a_2, \ldots, a_{|B|})$$

tale che

$$g(r) = p_1^{a_1} p_2^{a_2} \cdots p_{|B|}^{a_{|B|}}, \quad \text{con } \; p_i \in B.$$

Consideriamo poi la riduzione $w(r)$ di $v(r)$ modulo 2. Abbiamo k vettori in $(\mathbb{Z}/2\mathbb{Z})^{|B|}$, e poichè $k > B$ essi sono linearmente dipndenti. Con l'algoritmo di eliminazione di Gauss, che in $\mathbb{Z}/2\mathbb{Z}$ è particolarmente efficiente, si trovano $w(r_1), \ldots, w(r_t)$ tali che $w(r_1) + \cdots + w(r_t)$ sia il vettore nullo in $(\mathbb{Z}/2\mathbb{Z})^{|B|}$.

Segue che $g(r_1) \cdots g(r_t)$ è un quadrato perfetto y^2, e

$$g(r_1) \cdots g(r_t) \equiv r_1^2 r_2^2 \cdots r_t^2 \mod n.$$

Abbiamo così una congruenza del tipo cercato.

C'è circa il 50% di probabilità di ottenere un fattore proprio di n, ma se $k > |B|+10$, possiamo trovare almeno 10 relazioni di dipendenza lineare, provare con queste.

Occupiamoci ora delle pecche di questo algoritmo, ad esempio nel caso di Dixon. Se n è un intero di 25 cifre, anche la maggior parte dei numeri $g(r)$ avranno 25 cifre, e la probabilità che $g(r)$ abbia tutti i fattori primi minori di 10 000 è di $\frac{1}{50\,000}$, il che significa che occorre scgliere a caso 62 milioni di valori di r in modo da ottenere 1230 valori $g(r)$ che si fattorizzino completamente (si ricordi che in questo caso $|B| = 1229$). In realtà, se scegliamo r vicno a $\sqrt{n}$, allora $g(r)$ sarà vicino a $2\sqrt{n}$, e serve fare circa un milone di tentativi.

È per risolvere questo problema che entra in gioco l'idea di Pomerance.

6.8 Il Crivello Quadratico

Carl Pomerance ha migliorato notevolmente l'algoritmo di Dixon introducendo un sistema di crivello, che permette di stabilire quali numeri sono divisibili per i primi p nella base di fattori senza dover eseguire la divisione.

Si parte dall'insieme B di tutti i primo minori o uguali a P, ove P è un opportuno intero. Si supponga anche di aver verificato che n non è divisibile per nessun primo in B.

Anziche scegliere gli r a caso si considerano tutti gli interi r tali che $\lceil\sqrt{n}\rceil \leq r \leq \lceil\sqrt{n}\rceil + A$, ove A è scelto in modo opportuno, e si abbia anche che $\lfloor\sqrt{n}\rfloor + A \leq \sqrt{2n}$. Si ha allora che $g(r) = f(r)$, ove $f(r) = r^2 - n$.

Si ha che se un primo p di B divide $f(r)$, allora $r^2 \equiv n \bmod p$, quindi n è un residuo quadratico modulo p. In particolare basta controllare che il simbolo di Legendre $\left(\frac{n}{p}\right)$ sia 1, e per far ciò esiste un algoritmo molto efficiente (si veda [1, Chapter 5]). Si eliminano quindi dalla base di fattori tutti i primi che non hanno questa proprietà.

Il numero P deve essere abbastanza grande da avere in B un numero sufficiente di primi per fattorizzare parecchi $f(r)$, ma non troppo per non dover applicare l'algoritmo di Gauss a vettori $w(r)$ troppo lunghi. Tipicamente si scelgono P ed A dell'ordine di grandezza di

$$e^{\sqrt{\log n \log\log n}},$$

e in modo che sia $P < A < P^2$.

Dobbiamo ora controllare quali $f(r)$ si fattorizzano completamente utilizzando i primi in B. Se n è un residuo quadratico modulo p, con p dispari, si ha che $n \equiv t^2 \bmod p$ per qualche t, quindi $r^2 \equiv t^2 \bmod p$, da cui segue $r \equiv t$ oppure $r \equiv -t \bmod p$. Quindi, una volta calcolato t, riusciamo a determinare senza eseguire nessuna divisione tutti e soli gli $f(r)$ che sono divisibili per p. Siano r_0 ed r_1 i primi due interi nell'intervallo $I = [\lceil\sqrt{n}\rceil, \lceil\sqrt{n}\rceil + A]$ tali che $r_0 \equiv t \bmod p$ e $r_1 \equiv -t \bmod p$. Allora gli $f(r)$ che ci interessano sono $f(r_i + p), f(r_i + 2p), \ldots f(r_i + jp), \ldots$, per $i = 1, 2$. In questo modo si esegue la procedura di crivello.

Questa è l'idea di base. In realtà bisogna essere un pò più precisi.

Costruiamo una matrice in cui nella prima colonna scriviamo i valori di r, nella seconda colonna i valori di $f(r) = r^2 - n$, poi mettiamo tante altre colonne corrispondenti ai primi di B.

Come lavoriamo sulla colonna corrispondente al primo numero primo $p = p_1$ di B? Supponiamo per ora p dispari.

Ci interessa conoscere la massima potenza di p, diciamo p^β, che divide $f(r)$. Si ha che $p^\beta | r^2 - n$, quindi esiste t tale che $n \equiv t^2 \bmod p^\beta$. Si parte con $\alpha = 1$ e si cercano soluzioni della congruenza

$$n \equiv t^2 \mod p^\alpha \tag{6.2}$$

che siano comprese nell'intervallo I, poi si pone $\alpha = 2$ e di nuovo si cercano soluzioni in I della congruenza 6.2, e così via. Ad un certo punto si trova α tale

che esistono soluzioni in I della congruenza 6.2, ma non esistono soluzioni in I della congruenza $n \equiv t^2 \bmod p^{\alpha+1}$ e si pone $\beta = \alpha$. Siano $t_1, t_2 \in I$ soluzioni di 6.2 con $\alpha = \beta$, tali che $t_1 \equiv t_2 \bmod p^\beta$.

- $*$ Nella colonna corrispondente a p si segna un β in corrispondenza di t_1, e si tiene nota di $f(t_1)$ diviso p^β (o meglio, per evitare di eseguire divisioni, di $\log f(t_1) - \beta \log p$), poi si segna un altro β in corrispondenza di $t_1 + p^\beta$, e si tiene nota di $f(t_1 + p^\beta)$ diviso p^β (o meglio di $\log f(t_1 + p^\beta) - \beta \log p$). Analogamente per $t_1 + 2p^\beta$, $t_1 - p^\beta$, $t_1 - 2p^\beta$ e così via, procedendo sia verso l'alto che verso il basso a passi di p^β. Poi si considera $t_1 + p^{\beta-1}$ e si segna in corrispondenza di esso un $\beta - 1$, tenendo nota di $f(t_1 + p^{\beta-1})$ diviso $p^{\beta-1}$, poi si procede con $t_1 + 2p^{\beta-1}$ e così via, segnando $\beta - 1$ in corrispondenza di $f(t_1 + sp^{\beta-1})$ e tenendo nota di $f(t_1 + sp^{\beta-1})$ diviso $p^{\beta-1}$ (purchè la casella di coordinate $(t_1 + sp^{\beta-1}, p)$ non fosse già occupata, nel qual caso non si fa nulla). Poi si considera $\beta - 2$ e di nuovo si procede con la stessa tecnica a partire da $t_1 + p^{\beta-1}$, sia verso l'alto che verso il basso a passi di $\beta - 2$. L'ultima volta che si esegue il crivello sulla colonna corrispondente a p è a passi di $p = p^1$.

Si ripete poi il passo $*$ con t_2 al posto di t_1.

Questo per p_1. Analogo discorso per p_2, solo che questa volta in qualche caso avremo già diviso $f(r)$ per qualche potenza di p_1, e prenderemo nota di $f(r)/p_1^{\alpha_1}$ diviso $p_2^{\alpha_2}$, per opportuni α_1 e α_2 (o meglio di $\log f(r) - \alpha_1 \log p_1 - \alpha_2 \log p_2$).

Si procede in questo modo per tutti i primi dispari in B. Vediamo ora la procedura per $p = 2$.

Per il primo 2 la procedura è simile ma con qualche accortezza in più (e a dire il vero tipicamente si inizierebbe proprio con $p_1 = 2$). Osserviamo che se $n \equiv 3$ o $n \equiv 7 \bmod 8$ allora per r dispari $f(r)$ è divisibile per 2 ma non per 4; se $n \equiv 5 \bmod 8$ allora per r dispari $f(r)$ è divisibile per 4 ma non per 8; se $n \equiv 1 \bmod 8$ allora per r dispari $f(r)$ è divisibile per p^α con $\alpha \geq 3$ e si trova β in modo analogo al caso per p dispari. Anche il passo $[*]$ è analogo a quello per p dispari.

Osserviamo anche che è ininfluente l'ordine con cui ci si occupa delle colonne corrispondenti ai vari primi p di B. In questa esposizione abbiamo trattato il caso $p = 2$ alla fine, in pratica però ci se ne occupa all'inizio.

A questo punto le righe in cui $f(r)$, dopo le divisioni per tutti i primi di B, è diventato 1 (o meglio il logaritmo è all'incirca zero, dove "all'incirca" dipende dalla precisione con cui viene approssimato il logaritmo) corrispondono agli $f(r)$ che si fattorizzano completamente con i primi di B, e da tali righe deduciamo immediatamente il vettore $v(r)$ associato ad r. Si procede poi con il metodo di eliminazione di Gauss come per l'algoritmo di Dixon.

Il vantaggio di questo algorimo è che, oltre ad essere più efficiente, la parte del crivello può essere eseguita da più calcolatori in parallelo.

6.9 Fattorizzazione con primi grandi e polinomi multipli

Presentiamo ora due miglioramenti al crivello quadratico. Il primo riguarda la fattorizzazione con primi grandi.

Dopo aver effettuato la procedura di crivello, si consideravano esclusivamente gli $f(r)$ che si fattorizzavano completamente con i primi di B, cioè quelli che, dopo aver effettuato le divisioni (o meglio le sottrazioni, utilizzando i logaritmi) per tutti i possibili primi in B, davano come quoziente rimanente 1.

È possibile che ci siano vari $f(r)$ che danno come quoziente rimanente uno stesso primo grande q non appartenente a B. Il prodotto di qualsiasi di tali $f(r)$, diciamo $f(r_{q,i})$ e $f(r_{q,j})$ sarà il prodotto di primi in B per q^2, e per i nostri scopi questo va altrettanto bene che trovare una fattorizzazione completa con primi in B, visto che l'obiettivo è quello di trovare un prodotto di certi $f(r)$ che sia un quadrato perfetto.

Questa situazione si può verificare per vari primi grandi $q_1, \ldots, q_s$. In tal caso si considerano i vettori $v(r)$ di prima, cui si aggiungono s componenti tutte uguali a zero in fondo (corrispondenti ai nuovi primi), e si considerano anche i nuovi vettori $v(r_{q_h,i}) + v(r_{q_h,j})$, dove nel posto corrispondente a q_h compare un 2, che indica che q_h^2 divide $f(r_{q_h,i})f(r_{q_h,j})$.

Il secondo miglioramento fu suggerito da Peter Montgomery. Più gli $f(r)$ sono piccoli, più è probabile che si fattorizzino completamente, quando r è vicino a $\sqrt{n}$.

Si sceglie un intervallo I del tipo $I = [\sqrt{n} - M, \sqrt{n} + M]$, e anzichè utilizzare il polinomio $f(r) = r^2 - n$, si considerano polinomi del tipo $F(r) = ar^2 + 2br + c$, ove a, b, c sono scelti in modo che a sia primo e $n = b^2 - ac$.

Si ha che $aF(r)) = (ar + b)^2 - n$, quindi $F(r)$ ha ancora la proprietà che se un primo p divide $F(r)$, allora n è un residuo quadratico modulo p, e si può procedere come per il crivello quadratico. Si noti che ora $F(r)$ può anche essere negativo, ma basta aggiungere -1 alla base di fattori.

Si noti che la funzione $F(r)$ è una parabola che ha il punto di minimo in $r = -b/a$. Scegliamo come intervallo I in cui effettuare il crivello l'intervallo $I = [-b/a - M, -b/a + M]$. Per fare in modo che i valori assunti da $F(r)$ siano piccoli, imponiamo che sia $|F(-b/a)| = |F(-b/a \pm M)|$, cioè $n/a = aM^2 - n/a$, da cui segue che a è un primo della grandezza di $\frac{\sqrt{2n}}{M}$.

Si prende poi come b una soluzione della congruenza $x^2 \equiv n \bmod a$, e infine $c = (b^2 - n)/a$.

La scelta di questo polinomio fa sì che gli $F(r)$ siano più piccoli dei corrispondenti $f(r)$, quindi è più probabile che si fattorizzino. Questo ci permette di eseguire il crivello a partire da un intervallo I più piccolo, con un notevole risparmio di tempo.

Infine, per una veloce descrizione del crivello con i campi di numeri si veda [4, p.164] oppure [6, p.134].

Bibliografia

[1] Bressoud, David M, Factorization and primality testing. Undergraduate Texts in Mathematics. Springer-Verlag, New York, 1989.

[2] Buchmann, Johannes A. Introduction to cryptography. Undergraduate Texts in Mathematics. Springer-Verlag, New York, 2001.

[3] Giblin Peter, Primes and Programming, Cambridge University press, Cambridge, 1993.

[4] Koblitz, Neal, A course in number theory and cryptography. Graduate Texts in Mathematics, 114. Springer-Verlag, New York, 1987.

[5] Kranakis, Evangelos Primality and cryptography. Wiley-Teubner Series in Computer Science. John Wiley & Sons, Ltd., Chichester; B. G. Teubner, Stuttgart, 1986.

[6] Languasco Alessandro, Zaccagnini Alessandro, Introduzione alla crittografia moderna, 2003.

[7] Hernstein, Algebra, Editori Riuniti, Roma, 1982.

[8] Jacobson, Nathan Basic algebra. I. Second edition. W. H. Freeman and Company, New York, 1985.

[9] Mattarei Sandro, Note di Teoria dei Numeri e Crittografia, http://www-math.science.unitn.it/~mattarei/

[10] Menezes, Alfred J.; van Oorschot, Paul C.; Vanstone, Scott A., Handbook of applied cryptography. With a foreword by Ronald L. Rivest. CRC Press Series on Discrete Mathematics and its Applications. CRC Press, Boca Raton, FL, 1997.

[11] Riesel Hans, "Prime Numbers and Computer Methods for Factorization, Birkhäuser, Boston, 1994.