UNIVERSITA' DEGLI STUDI DI CATANIA
Facoltà di Scienze Matematiche, Fisiche e Naturali
Corso di Laurea in Informatica

Test di Primalità Probabilistici e Deterministici

Tesi di Laurea di
Giancarlo D' Urso

Relatore : Chiar.mo Prof. Domenico Cantone

Anno Accademico 2006 - 2007

Ai miei genitori.

Ringraziamenti

Ringrazio il Professore Domenico Cantone, del Dipartimento di Matematica e Informatica, presso l'Università degli Studi di Catania, che mi ha guidato nella stesura della tesi, e aiutato in ogni mia difficoltà.

Inoltre, vorrei ringraziare il professore Burton Rosenberg, della Miami University, e il professor Hendrik Lenstra, della University of California, per aver instaurato gentilmente una corrispondenza elettronica volta al chiarimento di alcuni argomenti trattati.

Infine, dedico la tesi ai miei genitori, che in tutti questi anni di studio, mi hanno sostenuto materialmente e moralmente.

Indice

Simboli e Notazioni

$\mathbb{N}$	insieme dei numeri naturali
$\mathbb{Z}$	insieme dei numeri interi
$\mathbb{R}$	insieme dei numeri reali
$\mathbb{C}$	insieme dei numeri complessi
$\mathbb{P}$	insieme dei numeri primi
$\mathbb{N}\backslash\mathbb{P}$	insieme dei numeri composti
$\mathbb{Z}_n$	gruppo additivo modulo n
$\mathbb{Z}_n^*$	gruppo moltiplicativo modulo n
$\mathbb{Z}_n[x]$	insieme dei polinomi a coefficienti in $\mathbb{Z}_n$
$\mathbb{Z}_n[x]/h(x)$	anello di polinomi modulo $h(x)$ in $\mathbb{Z}_n[x]$
$\ln n$	logaritmo naturale di n
$\log n$	logaritmo in base due di n
$\lfloor x \rfloor$	intero più grande, minore o uguale a x
$\lceil x \rceil$	intero più piccolo, maggiore o uguale a x
$a \mid b$	a divide b
$a \nmid b$	a non divide b
$\sqrt{x}$	radice quadrata di x
a^b	elevamento a potenza
$n!$	fattoriale di n
e	numero di Nepero
i	unità immaginaria

π	pigreco
$\phi(n)$	funzione totiente
$\gcd(a, b)$	massimo comune divisore tra a e b
$lcm(a, b)$	minimo comune multiplo tra a e b
$\sin(x)$	seno di x
$\cos(x)$	coseno di x
$\int_a^b f(x)dx$	integrale definito tra a e b di $f(x)$ in x
$\sum$	simbolo di sommatoria
$\prod$	simbolo di produttoria
$\lim_{x\to\infty} f(x)$	limite per x che tende ad infinito di $f(x)$
$ord_r(n)$	ordine di n modulo r
$\nu_p(n)$	massimo esponente di una potenza in base p che divide n
$\left(\frac{a}{b}\right)$	simbolo di Jacobi e/o Legendre
$\binom{a}{b}$	coefficiente binomiale
$Pr[E]$	probabilità dell'evento E
$\mathcal{O}$	simbolo di O-grande
$\oplus$	operazione generica
$a \equiv b \pmod{n}$	a congruo b modulo n
$a \not\equiv b \pmod{n}$	a non è congruo b modulo n
$f(x) \equiv g(x) \pmod{p}$	coefficienti di $f(x)$ e $g(x)$ congrui modulo p
$f(x) \equiv g(x) \pmod{h(x), p}$	$f(x) \equiv g(x) \pmod{h(x)}$ in $\mathbb{Z}_p[x]$

Prefazione

La matematica si è sempre interessata allo studio dei numeri primi, cercando di evidenziarne le proprietà e accentuarne le differenze con gli altri numeri interi, detti composti. Oggi, molte di queste caratteristiche sono ancora da formalizzare e rimangono delle congetture. Recentemente è stato risolto un problema, noto in letteratura come "PRIMES"e che consiste nel trovare un algoritmo efficiente per testare la primalità di un numero. In realtà, l'interesse sull'efficienza computazionale si è diffuso solo negli anni sessanta con l'avvento dei calcolatori elettronici, ma da parecchi secoli, i matematici si sono cimentati nella produzione di algoritmi di primalità, più o meno veloci. Il più antico risale al 240 a.C. ed è il Crivello di Eratostene. Questo algoritmo produce una lista di numeri primi minori o uguali ad un intero scelto, ma con complessità esponenziale rispetto alla dimensione dell'input. Il XVII secolo segna un passo in avanti nella ricerca, grazie al piccolo teorema di Fermat. Il teorema dimostrò una proprietà per tutti i numeri primi, valida anche per alcuni numeri composti (detti di Carmichael), fornendo lo strumento teorico per costruire il primo algoritmo probabilistico della storia: il test di pseudoprimalità di Fermat.

Nel 1975, Diffie e Hellman presentarono una nuova teoria crittografica, basata sui crittosistemi a chiave pubblica, che anzichè usare una stessa chiave (mantenuta segreta) tra due interlocutori per codificare e decodificare le informazioni, prevedeva due chiavi distinte, una per la codifica (chiave pubblica) e una per la decodifica (chiave privata e segreta). La chiave privata non poteva essere calcolata "facilmente"a partire dalla chiave pubblica, pertanto solo il possessore della chiave privata era in grado di decodificare le

informazioni. L'applicazione pratica di questa teoria incentivò l'uso dei numeri primi, basti pensare che la robustezza dell'RSA trae origine dalla difficoltà nel fattorizzare un numero, di cui si sa essere il prodotto di due numeri primi. Quindi, stabilire se un numero è primo o composto, è utile per la crittografia, quanto per la crittoanalisi. Più la risposta di un algoritmo di primalità è veloce, più l'RSA e altri crittosistemi diventano vulnerabili. Così, nello stesso periodo, nascono algoritmi probabilistici polinomiali come l'algoritmo Solovay-Strassen (1975) e l'algoritmo Miller-Rabin (1976).

Gli ultimi decenni vedono il proliferarsi di algoritmi sempre più veloci, tanto da far avanzare l'ipotesi che si possa costruire un algoritmo deterministico e contemporaneamente polinomiale. Il traguardo è sempre più vicino nel 1983, quando Adleman, Pomerance e Rumely svilupparono un algoritmo deterministico di complessità $(\log n)^{\mathcal{O}(\log\log\log n)}$. Ma nel 2002, tre matematici indiani (Agrawal, Kayal, Saxena) raggiungono l'obiettivo diffondendo un articolo "PRIMES is in P"destinato a rimanere nella storia e che darà origine a una sequenza di miglioramenti, fino a raggiungimento della forma attuale (6° versione).

Questa tesi illustra attraverso una scelta mirata di algoritmi, l'evoluzione storica del problema, mostrando i progressi che sono stati raggiunti in termini di complessità e di fattibilità realizzativa. La trattazione si sofferma anche sulla differenza tra algoritmi deterministisci (cioè algoritmi che producono output sicuramente corretti) e algoritmi probabilistici (cioè algoritmi che possono, per loro natura, produrre output errati ma con basse probabilità di errore). Per ottemperare agli obiettivi previsti, la tesi si articola in sei capitoli. I capitoli iniziali (primo e secondo) contengono tutti gli stumenti matematici per la comprensione degli algoritmi trattati, tra i quali ricordiamo il teorema cinese del resto, il piccolo teorema di Fermat, la legge di reciprocità quadratica, la teoria sui polinomi ciclotomici e risultati di teoria analitica dei numeri. I quattro capitoli successivi, trattano in sequenza il crivello di Eratostene, l'algoritmo di Miller-Rabin, l'algoritmo Solovay-Strassen e l'algoritmo Agrawal-Kayal-Saxena. Per ciascun algoritmo è stata dimostrata la correttezza e la complessità, mentre per gli algoritmi probabilistici è previsto un paragrafo che ne descrive la probabilità di errore. Infine, il lavoro è corredato

da una implementazione in java degli ultimi tre algoritmi, mediante la quale è possibile confrontare i risultati ottenuti e i rispettivi tempi di elaborazione, per una data lista di numeri.

Parte I

Preliminari di algebra e di teoria dei numeri

Capitolo 1

Richiami di Algebra

1.1 Numeri interi e congruenze

Per trattare il problema della primalità dei numeri, è necessario studiare l'insieme dei numeri naturali e l'insieme degli interi relativi. Purtroppo l'introduzione di questi insiemi nella cultura matematica e la loro trattazione completa e rigorosa non è banale. Mi limiterò dunque ad esporre un elenco minimale di proprietà di $\mathbb{Z}$ ed $\mathbb{N}$.
Inizio, dunque, introducendo il concetto di divisione.

Teorema 1 (Proprietà euclidea degli interi)**.** *Siano* $a, b \in \mathbb{Z}$ *e* $a \neq 0$*. Allora* $\exists q \in \mathbb{Z}$ *e* $\exists r \in \mathbb{N}$ *tale che* $b = qa + r$ *con* $0 \leq r <\mid a \mid$.

Se $b = qa$, per qualche q, si dice che a divide b (o che a è un divisore di b, o che b è un multiplo di a). A volte per indicare ciò, scriviamo $a \mid b$. Mentre scriviamo $a \nmid b$ per indicare che a non divide b.

Enunciamo qualche proprietà dei divisori comuni.

1. $\forall a \in \mathbb{Z},\ \ a \mid a$
2. $\forall a, b, c \in \mathbb{Z},\ \ a \mid b$ e $b \mid c \Rightarrow a \mid c$
3. $\forall a, b, d \in \mathbb{Z},\ \ d \mid a$ e $d \mid b \Rightarrow d \mid (a + b)$ e $d \mid (a - b)$
4. $\forall a, b, d, x, y \in \mathbb{Z},\ \ d \mid a$ e $d \mid b \Rightarrow d \mid (ax + by)$
5. $\forall a, b \in \mathbb{Z},\ \ a \mid b$ e $b \mid a \Rightarrow a \in \{+b, -b\}$

Definizione 1 (Massimo comune divisore)**.** *Siano* $a, b \in \mathbb{Z}$ *con* $a \neq 0$ *e* $b \neq 0$*. Definiamo* massimo comune divisore *tra* a *e* b *e lo denotiamo con* $\gcd(a, b)$ *quel numero* $h \in \mathbb{N}$ *tale che :*

1. $\forall d \in \mathbb{Z}$ *se* $d \mid a$ *e* $d \mid b$ *allora* $d \mid h$
2. $\exists x, y \in \mathbb{Z} : h = ax + by$

Il massimo comune divisore di due interi a e b, è il più grande dei divisori comuni di a e b. Quando $a = b = 0$, per convenzione, si assume $\gcd(0, 0) = 0$. Inoltre se a e b non sono entrambi nulli, allora $1 \leq \gcd(a, b) \leq \min(\mid a \mid, \mid b \mid)$. Quelle che seguono, sono le proprietà elementari della funzione gcd :

1. $\forall a, b \in \mathbb{Z},\ \gcd(a, b) = \gcd(b, a)$
2. $\forall a, b \in \mathbb{Z},\ \gcd(a, b) = \gcd(-a, b)$
3. $\forall a, b \in \mathbb{Z},\ \gcd(a, b) = \gcd(\mid a \mid, \mid b \mid)$
4. $\forall a \in \mathbb{Z},\ \gcd(a, 0) = \mid a \mid$
5. $\forall a, k \in \mathbb{Z},\ \gcd(a, ka) = \mid a \mid$
6. $\forall a, b \in \mathbb{Z}$ e $n \in \mathbb{N},\ \gcd(na, nb) = n \gcd(a, b)$

Teorema 2. *Se $a, b \in \mathbb{Z}$, non entrambi nulli, allora* $\gcd(a, b)$ *è il più piccolo elemento positivo dell'insieme* $\{ax + by : x, y \in \mathbb{Z}\}$.

Dimostrazione. Sia s il più piccolo positivo intero dell'insieme $\{ax + by : x, y \in \mathbb{Z}\}$ e si abbia $s = ax_0 + by_0$. Sia $q = \lfloor a/s \rfloor$. Sappiamo che :

$$a \mod s = a - qs = a - q(ax_0 + by_0) = a(1 - qx_0) + b(-qy_0).$$

Dunque anche $a \mod s$ sta nell'insieme delle combinazioni lineari. Ma, poichè $a \mod s < s$ si ha che $a \mod s = 0$. Allora, $s \mid a$ e per lo stesso motivo $s \mid b$. Pertanto s è un divisore comune di a e di b. Quindi $\gcd(a, b) \geq s$, perchè s è una combinazione lineare di a e b. Ma per la proprietà 4 dei divisori comuni, $\gcd(a, b) \mid s$. Ciò implica che $\gcd(a, b) \leq s$. In definitiva abbiamo che $\gcd(a, b) = s$ e il teorema è dimostrato. □

Generalmente, il massimo comune divisore tra due numeri interi si calcola utilizzando la scomposizione in fattori primi degli operandi (Definizione 7). In particolare si moltiplicano tutti i fattori primi comuni tra i due numeri, elevandoli al minimo esponente con cui compaiono nella scomposizione in fattori.

Questa procedura, da un punto di vista computazionale, è troppo onerosa. Nelle applicazioni, si adotta un algoritmo diverso, più efficiente e di particolare rilevanza storica. Si tratta dell'algoritmo di Euclide. L'algoritmo è il seguente :

euclide(a, b)

1. if $(b == 0)$ then
2. return a;
3. else
4. return **euclide**$(b, a \mod b)$;

La correttezza di questo algoritmo può essere dimostrata con il seguente teorema.

Teorema 3. $\forall a, b \in \mathbb{Z}^+,\ \gcd(a, b) = \gcd(b, a \mod b)$.

Dimostrazione. Per dimostrare che $\gcd(a, b) = \gcd(b, a \mod b)$ basta dimostrare che $\gcd(a, b) \mid \gcd(b, a \mod b)$ e $\gcd(b, a \mod b) \mid \gcd(a, b)$. Mostriamo che $\gcd(a, b) \mid \gcd(b, a \mod b)$. Poniamo $h = \gcd(a, b)$ e $(a \mod b) = a - \lfloor a/b \rfloor b$. Poichè $(a \mod b)$ è una combinazione lineare di a e b, si ha che $h \mid (a \mod b)$. E quindi poichè $h \mid b$ concludiamo che $h \mid \gcd(b, a \mod b)$.
Viceversa, per dimostrare che $\gcd(b, a \mod b) \mid \gcd(a, b)$, se poniamo $h = \gcd(b, a \mod b)$, allora $h \mid b$ e $h \mid (a \mod b)$. Poichè $a = \lfloor a/b \rfloor b + (a \mod b)$, si ha che a è combinazione lineare di b e di $(a \mod b)$. E quindi si conclude che $h \mid a$, da cui segue la tesi. □

Infine, per dimostrare che l'algoritmo di Euclide non richiama se stesso all'infinito, basta notare che ad ogni chiamata ricorsiva, il secondo argomento decresce. Infatti la quantità $a \mod b$ è sempre minore di b e maggiore o uguale di zero. Quindi il decremento non può andare avanti all'infinito e termina proprio quando il secondo argomento è zero. In tal caso, l'algoritmo restituisce il primo argomento applicando la proprietà $\gcd(a, 0) = a$.

Per determinare la complessità dell'algoritmo, è importante stabilire il numero di chiamate ricorsive richieste per portarlo a termine. Il teorema di Lamè, che nel suo enunciato fa uso dei numeri di Fibonacci, ci fornisce una soluzione.

Definizione 2 (numeri di Fibonacci)**.** *I numeri di Fibonacci sono definiti dalla seguente ricorrenza :*

$$F_0 = 0$$

$$F_1 = 1$$

$$F_i = F_{i-1} + F_{i-2},\ \forall i \geq 2.$$

Teorema 4 (Lamè)**.** *Se $a > b \geq 0$ e l'algoritmo di Euclide applicato alla coppia (a, b), esegue $k \geq 1$ iterazioni, allora $a \geq F_{k+2}$ e $b \geq F_{k+1}$.*

Dimostrazione. Dimostriamo il teorema per induzione, sul numero k di chiamate ricorsive

Caso base : $k = 1$. Poichè viene eseguita una chiamata ricorsiva, si ha che $b \neq 0$. Quindi $b \geq 1 = F_2$. Inoltre, poichè $a > b$ si ha che $a \geq 2 = F_3$.
Il caso base può verificarsi dopo una serie di chiamate ricorsive. In tutte queste chiamate b sarà sempre maggiore di $(a \mod b)$, quindi la condizione $a > b$ varrà sempre. Questo è fondamentale per la verifica del caso base.

Passo induttivo : Supponiamo il teorema vero per $k - 1$ chiamate ricorsive, e dimostriamolo per k chiamate ricorsive. Cioè supponiamo che se $euclide(b, a \mod b)$ esegue $k - 1$ iterazioni allora $b \geq F_{k+1}$ e $b \geq F_k$, e dimostriamo che $euclide(a, b)$ è tale da soddisfare la condizione $a \geq F_{k+2}$. Poichè $a > b$ (in qualsiasi chiamata ricorsiva) avremo che $\lfloor a/b \rfloor \geq 1$, da cui:

$$a \geq b + (a - \lfloor a/b \rfloor b) = b + (a \mod b) \geq F_{k+1} + F_k = F_{k+2}.$$

□

Grazie al teorema di Lamè, sappiamo che $F_{k+2} \leq a$ e poichè, come è ben noto, $F_i \simeq \left(\frac{1+\sqrt{5}}{2}\right)^i$, si ha :

$$F_{k+2} \leq a \Leftrightarrow \left(\frac{1+\sqrt{5}}{2}\right)^{k+2} \leq a \Leftrightarrow k \leq \left(\log_{\left(\frac{1+\sqrt{5}}{2}\right)} a\right) - 2$$

Quindi il numero di chiamate ricorsive è $\mathcal{O}(\log a)$. Ad ogni chiamata ricorsiva, si effettua una riduzione modulo b e $b < a = 2^{\log a}$. Ciascuna di queste riduzioni modulari richiede al più $\mathcal{O}((\log a)^2)$. Concludiamo che la complessità dell'algoritmo di Euclide è $\mathcal{O}((\log a)^3)$.

Definizione 3 (minimo comune multiplo)**.** *Siano $a, b \in \mathbb{N}^+$. Definiamo* minimo comune multiplo *tra a e b, e lo denotiamo con $lcm(a, b)$, l'intero positivo f tale che:*

1. *$a \mid f$ e $b \mid f$*
2. *$\forall c \in \mathbb{N}^+$, se $a \mid c$ e $b \mid c$, allora $f \mid c$*

Di seguito, riportiamo le proprietà elementari della funzione lcm.

1. $\forall a, b \in \mathbb{N}^+$, se $a = \prod_{i=1}^{m} p_i^{\alpha_i}$ e $b = \prod_{i=1}^{m} p_i^{\beta_i}$ sono la fattorizzazione canonica (vedi Definizione 7) di a e b, allora $lcm(a, b) = \prod_{i=1}^{m} p_i^{\max\{\alpha_i, \beta_i\}}$
2. $\forall a, b \in \mathbb{N}^+$, $lcm(a, b) = lcm(b, a)$
3. $\forall a, b, c, m \in \mathbb{N}^+$, $lcm(ma, mb, mc) = m \cdot lcm(a, b, c)$
4. $\forall a, b, c \in \mathbb{N}^+$, $lcm(a, b, c) = lcm(lcm(a, b), c) = lcm(a, lcm(b, c))$
5. $\forall a, b \in \mathbb{N}^+$, $lcm(a, b) = \frac{a \cdot b}{\gcd(a,b)}$

Definizione 4. *Due interi a,b si dicono* primi (o coprimi) tra di loro *se e solo se* $\gcd(a, b) = 1$.

Definizione 5 (numeri primi)**.** *Un numero intero p si dice* numero primo *se $p > 1$ e gli unici divisori di p sono $1, -1, p, -p$.*

Un intero $n > 1$ che non è primo è detto *numero composto*. Denotiamo con $\mathbb{P}$ l'insieme dei numeri primi e con $\mathbb{N} \backslash \mathbb{P}$ l'insieme dei numeri composti. Adesso dimostriamo che i numeri primi sono infiniti.

Lemma 1. *Se $n \in \mathbb{N}$ e $n > 1$, allora esiste $p \in \mathbb{P}$ tale che $p \mid n$.*

Dimostrazione. Sia $n \in \mathbb{N}$ e $n > 1$. Allora l'insieme $S = \{d \in \mathbb{N} : d \mid n, d \neq 1\}$ non è vuoto e quindi ammette un minimo. Sia $p = \min S$. Chiaramente $p > 1$. Ora, sia d un divisore positivo di p diverso da 1. Allora $d \in S$. Per la minimalità di p segue che $d = p$. Pertanto $p \in \mathbb{P}$. □

Teorema 5 (Euclide)**.** *L'insieme $\mathbb{P}$ dei numeri primi è infinito*

Dimostrazione. Supponiamo per assurdo che esistono un numero finito di k numeri primi. Siano essi $p_1, p_2, ..., p_k$. Consideriamo il numero

$$n = p_1 p_2 \cdots p_k + 1$$

Per il lemma precedente, poichè $n > 1$, esiste $p \in \mathbb{P}$ tale che $p \mid n$. Il numero primo p deve essere uno dei numeri $p_1, p_2, ...p_k$. Pertanto $p \mid n - p_1 p_2 \cdots p_k$, cioè $p = 1$. Assurdo, perchè per definizione un numero primo è diverso da 1. □

Definizione 6. *Sia $m \in \mathbb{Z}$. Definiamo la seguente relazione su $\mathbb{Z}$ ponendo, per ogni $a, b \in \mathbb{Z}$,*

$$a \equiv b \pmod{m} \iff m \mid (a - b)$$

La relazione si dice congruenza modulo m e se $a, b \in \mathbb{Z}$ tali che $a \equiv b \pmod{m}$, si dice che a è congruo b modulo m.

Teorema 6. *Sia $m \in \mathbb{Z}$. Allora*

1 - $\forall a \in \mathbb{Z}$, $a \equiv a \pmod{m}$

2 - $\forall a, b \in \mathbb{Z}, a \equiv b \pmod{m} \Rightarrow b \equiv a \pmod{m}$

3 - $\forall a, b, c \in \mathbb{Z}, a \equiv b \pmod{m} \wedge b \equiv c \pmod{m} \Rightarrow a \equiv c \pmod{m}$

Dimostrazione. 1 - Sia $a \in \mathbb{Z}$. Poichè $m \mid a - a = 0$, si ha che $a \equiv a \pmod{m}$

2 - Siano $a, b \in \mathbb{Z}$ tali che $a \equiv b \pmod{m}$. Allora $m \mid (a - b)$ e, quindi $m \mid -(a - b)$ cioè $m \mid (b - a)$. Pertanto $b \equiv a \pmod{m}$.

3 - Siano $a, b, c \in \mathbb{Z}$ tali che $a \equiv b \pmod{m}$ e $b \equiv c \pmod{m}$. Allora $m \mid (a - b)$ e

$m \mid (b-c)$. Poichè m è un fattore comune per $(a-b)\,e\,(b-c)$ si ha che $m \mid (a-b)+(b-c)$. Quindi $m \mid (a-c)$. Segue la tesi. □

Teorema 7. *Sia $m \in \mathbb{Z}$. Per ogni $a, b, c, d \in \mathbb{Z}$, se $a \equiv b \pmod m$ e $c \equiv d \pmod m$ allora*

$$a + c \equiv b + d \pmod m \text{ e } ac \equiv ad \pmod m$$

Dimostrazione. Siano $a, b, c, d \in \mathbb{Z}$ tali che $a \equiv b \pmod m$ e $c \equiv d \pmod m$. Allora $m \mid (a-b)$ e $m \mid (c-d)$. Quindi, essendo $(a-b)+(c-d) = (a+c)-(b+d)$, si ha che $m \mid (a+c)-(b+d)$. Pertanto $a+c \equiv b+d \pmod m$. Inoltre, se $m \mid (a-b)$ e $m \mid (c-d)$ si ha che :

$$m \mid a(c-d) + d(a-b) = ac - ad + ad - bd = ac - bd$$

Segue la tesi. □

Lemma 2. $\forall a, b, c \in \mathbb{Z}\ \ \gcd(a,c) = 1\ e\ \gcd(b,c) = 1 \Leftrightarrow \gcd(ab,c) = 1.$

Dimostrazione. Necessità : Dal Teorema 2 segue che esistono degli interi $x, y, x'y'$ tali che $ax + cy = 1$ e $bx' + cy' = 1$. Moltiplicando, membro a membro, queste due equazioni otteniamo $ab(xx') + c(ybx' + y'ax + cyy') = 1$. Poichè 1 è una combinazione lineare di ab e c , applicando il Teorema 2 otteniamo la tesi.

Sufficienza : Se $\gcd(ab,c) = 1$, dal Teorema 2, si ha che esistono $\overline{x}, \overline{y} \in \mathbb{Z}$ tali che $ab\overline{x} + c\overline{y} = 1$. Ciò equivale a scrivere che $a(b\overline{x}) + c\overline{y} = 1$ e $b(a\overline{x}) + c\overline{y} = 1$. E quindi per il Teorema 2, segue che $\gcd(a,c) = 1$ e $\gcd(b,c) = 1$. □

Teorema 8. *Sia $p \in \mathbb{P}$ e $a, b \in \mathbb{Z}$. Se $p \mid ab$ allora $p \mid a$ o $p \mid b$.*

Dimostrazione. Supponiamo per assurdo che $p \mid ab$ ma che $p \nmid a$ e $p \nmid b$. Poichè gli unici divisori di p sono 1 e p, si ha che $\gcd(a,p) = \gcd(b,p) = 1$. Allora dal lemma precedente segue che $\gcd(ab,p) = 1$, che contraddice l'ipotesi $p \mid ab$. Infatti se $p \mid ab$ allora $\gcd(ab,p) = p \neq 1$. □

Corollario 1. *Se* $k \in \mathbb{N}$, $(a_1, a_2, ..., a_k) \in \mathbb{Z}^k$ *e* $p \in \mathbb{P}$ *tali che*

$$p \mid \prod_{j=1}^{k} a_j$$

allora $\exists j \in \{1, 2, ..., k\}$ *tale che* $p \mid a_j$.

Definizione 7. *Dato* $n \in \mathbb{N}^+$ *chiamiamo fattorizzazione canonica (o scomposizione in fattori) di* n *l'espressione :*

$$n = \prod_{i=1}^{k} p_i^{\alpha_i}$$

dove $p_i < p_j$ *se* $i < j, p_i \in \mathbb{P}$ *e* $\alpha_i \in \mathbb{N}^+$ *per* $i = 1, ..., k$. *Se* $n = 1$ *allora* $k = 0$ *e il prodotto è vuoto.*

Teorema 9 (Teorema fondamentale dell'aritmetica)**.** *Per ogni numero* $n \in \mathbb{N}^+$ *esiste la fattorizzazione canonica ed è unica.*

Dimostrazione. Iniziamo col dimostrare l'esistenza della fattorizzazione canonica. Definiamo il predicato:

$$P(n) = \{\exists k \in \mathbb{N}, \exists (p_1, ..., p_k) \in \mathbb{P}^k \exists (a_1, ..., a_k) \in \mathbb{N}^k : n = \prod_{i=1}^{k} p_j^{a_j} \text{ con } p_i < p_j \text{ se } 1 \leq i < j \leq k\}$$

e dimostriamo il terema per induzione su n.

Caso base : $n = 1$ si ha $k = 0$ e $1 = \prod \emptyset$ pertanto vale $P(1)$.

Passo induttivo : Sia $P(l)$ vera per ogni $l < n + 1$ e dimostriamo che vale $P(n + 1)$.

Per il Lemma 1 sappiamo che $\exists p \in \mathbb{P}$ tale che $p \mid n + 1$. Sia $l \in \mathbb{N}$ tale che $n + 1 = pl$. Allora $l \in \mathbb{N}, l < n + 1$ e quindi, per l'ipotesi induttiva, vale $P(l)$. Dove:

$$P(l) = \{\exists h \in \mathbb{N} \exists (p_1, ..., p_h) \in \mathbb{P}^h \exists (a_1, ..., a_h) \in \mathbb{N}^h : n = \prod_{i=1}^{h} p_j^{a_j} \text{ con } p_i < p_j \text{ se } 1 \leq i < j \leq h\}$$

Pertanto $n + 1 = p \cdot (\prod_{j=1}^{h} p_j^{a_j})$, cioè è vera $P(n + 1)$.

Adesso, dimostriamo l'unicità della fattorizzazione cioè se $k, l \in \mathbb{N}, (p_1, ..., p_k) \in \mathbb{P}^k, (q_1, ..., q_l) \in \mathbb{P}^l, (a_1, ..., a_k) \in \mathbb{N}^k, (b_1, ..., b_l) \in \mathbb{N}^l$ tali che :

$$\prod_{i=1}^{k} p_j^{a_j} = \prod_{j=1}^{l} q_j^{b_j} \text{ e } p_1 < p_2 < ..... < p_k,\ q_1 < q_2 < ..... < q_l$$

allora $k = l$ e, per ogni $1 \leq j \leq k$, si ha $p_j = q_j$ e $a_j = b_j$.

Procediamo per induzione su k.

Caso base : Sia $k = 0$. allora $\prod_{j=1}^{k} p_j = 1$ e quindi $1 = \prod_{j=1}^{l} q_j$. Pertanto $l = 0$ e si ha la tesi.

Passo induttivo : Sia $k \in \mathbb{N}$, tale che se $l \in \mathbb{N}, (p_1, ..., p_k) \in \mathbb{P}^k, (q_1, ..., q_l) \in \mathbb{P}^l, (a_1, ..., a_k) \in \mathbb{N}^k, (b_1, ..., b_l) \in \mathbb{N}^l$ e :

$$\prod_{j=1}^{k} p_j^{a_j} = \prod_{j=1}^{l} q_j^{b_j} \text{ e } p_1 < p_2 < ..... < p_k,\ q_1 < q_2 < ..... < q_l$$

allora $k = l$ e, per ogni $1 \leq j \leq k$, si ha $p_j = q_j$ e $a_j = b_j$.

Dimostriamo la tesi per k+1. Siano $p_1, ..., p_{k+1} \in \mathbb{P}, q_1, ..., q_{l'} \in \mathbb{P}, a_1, ..., a_{k+1}, b_1, ..., b_{l'} \in \mathbb{N}$ tali che

$$\prod_{j=1}^{k+1} p_j^{a_j} = \prod_{j=1}^{l'} q_j^{b_j} \text{ e } p_1 < p_2 < ..... < p_{k+1},\ q_1 < q_2 < ..... < q_{l'}$$

Allora $p_{k+1} \mid \prod_{j=1}^{l'} q_j^{b_j}$. Quindi $\exists j < l'$ tale che $p_{k+1} \mid q_j^{a_j}$ e quindi $p_{k+1} \mid q_j$. Ma $q_j \in \mathbb{P}$ e quindi $p_{k+1} = q_j$. Pertanto $p_{k+1} \leq q_{l'}$. Anche $q_{l'} \mid \prod_{j=1}^{k+1} p_j^{a_j}$ e quindi, analogamente, $q_{l'} \leq p_{k+1}$. Pertanto $q'_l = p_{k+1}$. Abbiamo così

$$\prod_{j=1}^{k} p_j^{a_j} = \prod_{j=1}^{l'-1} q_j^{b_j} \text{ e } p_1 < p_2 < ..... < p_k,\ q_1 < q_2 < ..... < q_{l'-1}$$

Applicando l'ipotesi induttiva otteniamo che $k = l' - 1$, $p_j = q_j$ e $a_j = b_j$ per ogni $1 \leq j \leq k$ e poichè $p_{k+1} = q_{l'}$, vale la tesi. □

Teorema 10. *Sia $p \in \mathbb{P}$. $\forall k \in \mathbb{N}$ e $1 \leq k \leq p-1$ si ha che $p \mid \binom{p}{k}$.*

Dimostrazione. Sia $k \in \mathbb{N}$ e $1 \leq k \leq p-1$. Dalla definizione di coefficiente binomiale si ha che:

$$p! = \binom{p}{k} k!(p-k)!.$$

Poichè $k \leq p-1$ e $k \geq 1$, si ha che $p \nmid k!$. Infatti se p dividesse $k!$, il prodotto di alcuni numeri interi minori di k sarebbe una potenza di p, il che è impossibile data la definizione di primo e dato che $k < p$.

Inoltre $\forall k : 1 \leq k \leq p-1$, si ha che $1 \leq p-k \leq p-1$ e quindi $p \nmid (p-k)!$. E allora, poichè $p \mid p!$ ma $p \nmid k!$ e $p \nmid (p-k)!$, per il Corollario 1, segue che $p \mid \binom{p}{k}$. □

Teorema 11 (piccolo teorema di Fermat)**.** *Se $p \in \mathbb{P}$, allora $\forall a \in \mathbb{Z}$, $a^p \equiv a \pmod{p}$. In particolare, se $p \nmid a$, allora $a^{p-1} \equiv 1 \,(mod\, p)$.*

Dimostrazione. Sia $p \in \mathbb{P}$. Dimostriamo che $\forall a \in \mathbb{N}, a^p \equiv a \pmod{p}$, per induzione su a.

1. Se $a = 0$, la tesi è vera.

2. Sia $a \in \mathbb{N}$, supponiamo che la tesi sia vera per a e dimostriamola per $a+1$. Noi sappiamo che :

$$(a+1)^p = \sum_{k=0}^{p} \binom{p}{k} a^k = a^p + \sum_{k=1}^{p-1} \binom{p}{k} a^k + 1$$

che si può scrivere anche nella forma :

$$(a+1)^p - (a^p + 1) = \sum_{k=1}^{p-1} \binom{p}{k} a^k.$$

Notiamo che la sommatoria al secondo membro è divisibile per p perchè sappiamo, per il teorema precedente, che $\forall k : 1 \leq k \leq p-1, p \mid \binom{p}{k}$ e quindi possiamo scrivere che :

$$(a+1)^p \equiv (a^p + 1) \pmod{p}$$

Applicando l'ipotesi induttiva, $a^p \equiv a \pmod{p}$ otteniamo:

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

Adesso, se $a \in \mathbb{Z} \backslash \mathbb{N}$, allora $(-a)^p \equiv -a \pmod{p}$.
Se $p \neq 2$, allora p è dispari e quindi $-a^p \equiv -a \pmod{p}$ e pertanto $a^p \equiv a \pmod{p}$.
Se $p = 2$, da $(-a)^2 \equiv -a \pmod{2}$ si ha che $a^2 \equiv -a \pmod{2}$. Ma $-a \equiv a \pmod{2}$ e così $a^2 \equiv a \pmod{2}$. Pertanto si ha la tesi.

Infine, se $p \in \mathbb{P}$ ed $a \in \mathbb{Z}$ tali che $p \nmid a$. Allora da $a^p \equiv a \,(mod\, p)$, segue che $p \mid a^p - a$, cioè $p \mid a(a^{p-1} - 1)$. Per il Corollario 1 si ha che $p \mid a^{p-1} - 1$, cioè $a^{p-1} \equiv 1 \pmod{p}$. □

1.2 Strutture Algebriche

Definizione 8 (di Gruppo)**.** *Un gruppo* $(G, \oplus)$ *è un insieme* G *con un'operazione binaria* $\oplus$ *definita su* G *per cui valgono le seguenti proprietà :*
- Chiusura: $\forall a, b \in G$*, si ha* $a \oplus b \in G$
- Associatività : $\forall a, b, c \in G$*, si ha* $(a \oplus b) \oplus c = a \oplus (b \oplus c)$
- Identità : esite un elemento e_G *(unità) tale che* $\forall a \in G$ *si ha* $e_G \oplus a = a \oplus e_G = a$
- Esistenza del reciproco di un elemento : $\forall a \in G$*, esiste un unico elemento* $b \in G$ *tale che* $a \oplus b = b \oplus a = e_G$*.*

Un gruppo $(G, \oplus)$ che soddisfa la legge commutativa $a \oplus b = b \oplus a \forall a, b \in G$, si chiama gruppo *commutativo* o *abeliano.* Se un gruppo $(G, \oplus)$ ha un numero di elementi finito (ovvero $\mid G \mid < \infty$), allora il gruppo si dice *finito.*

Definizione 9 (di sottogruppo)**.** *Se* $(G, \oplus)$ *è un gruppo,* $G' \subseteq G$*, e anche* $(G', \oplus)$ *è un gruppo, allora* $(G', \oplus)$ *è detto sottogruppo di* $(G, \oplus)$*.*

Teorema 12. *Se* $(G, \oplus)$ *è un gruppo finito e* G' *è un qualsiasi sottoinsieme non vuoto di* G*, chiuso rispetto a* $\oplus$*, cioè tale che* $a \oplus b \in G' \forall a, b \in G'$*, allora* $(G', \oplus)$ *è un sottogruppo di* $(G, \oplus)$*.*

Questo teorema fornisce una strategia per costruire un sottogruppo di un gruppo finito $(G, \oplus)$. Dato un elemento $a \in G$ si prendono tutti gli elementi generabili da a usando l'operazione del gruppo. Ognuno di questi elementi dunque sarà del tipo $a^{(k)} = \bigoplus_{i=1}^{k} a$. L'insieme generato da a e che denotiamo con $[a]$, definito come $[a] = \{a^{(k)} : k \geq 1\}$ è un sottogruppo di $(G, \oplus)$. Infatti, per ogni $k \geq 1, a^{(k)} \in G \, \forall k \geq 1$ e quindi $[a] \subseteq G$ e poichè l'associatività dell'operazione $\oplus$ implica che $a^{(i)} \oplus a^{(j)} = a^{(i+j)}$, per il teorema precedente, si ha che $[a]$ è sottogruppo di G. In questo caso si dice che a è un generatore di $[a]$.

Definizione 10. *Un gruppo* $(G, \oplus)$ *si dice* ciclico *se esiste un elemento* $g \in G$ *tale che* $\forall a \in G$ *esiste un intero* k *per il quale si ha* $a = g^{(k)} = \bigoplus_{i=1}^{k} g$*. L'elemento* g *è chiamato* generatore *del gruppo* G*.*

Teorema 13 (di Lagrange)**.** *Se $(G, \oplus)$ è un gruppo finito e $(G', \oplus)$ è un sottogruppo di $(G, \oplus)$, allora $\mid G' \mid$ è un divisore di $\mid G \mid$.*

Definizione 11 (ordine di un elemento)**.** *Sia $(G, \oplus)$ un gruppo con e elemento identità. Definiamo ordine di a, e lo indichiamo con $ord(a)$, il minimo intero $t > 0$ tale che $a^{(t)} = e_G$.*

Teorema 14. *Sia $(G, \oplus)$ un gruppo finito. $\forall a \in S$ si ha che $ord(a) = \mid [a] \mid$.*

Corollario 2. *Se $(G, \oplus)$ è un gruppo finito con identità e_G, allora $\forall a \in G$ vale $a^{|G|} = e_G$.*

Teorema 15. *Sia $(G, \oplus)$ un gruppo finito con identità e_G, e sia $a \in G$ e $t \in \mathbb{Z}$. Se $a^{(t)} = e_G$ allora $ord(a) \mid t$.*

Dimostrazione. Sia $t = q \cdot ord(a) + r$, con $0 \leq r < ord(a)$, tale che $a^{(t)} = e_G$ allora

$$e_G = a^{(t)} = a^{(q \cdot ord(a) + r)} = (a^{ord(a)})^{(q)} \oplus a^{(r)} = (e_G)^{(q)} \oplus a^{(r)} = a^{(r)}.$$

Dalla minimalità di $ord(a)$ segue che $r = 0$, e quindi $ord(a) \mid t$. □

Teorema 16. *Sia a un elemento di un gruppo abeliano G, tale che per qualche primo p e intero $n \geq 1$, si abbia $a^{(p^n)} = e_G$ e $a^{(p^{n-1})} \neq e_G$. Allora $ord(a) = p^n$.*

Dimostrazione. Se m è l'ordine di a, e $a^{(p^n)} = e_G$, allora $m \mid p^n$. Quindi $m = p^f$ per qualche $0 \leq f \leq n$. Se $f < n$, allora $a^{(p^{n-1})} = e_G$, che contraddice l'ipotesi $a^{(p^{n-1})} \neq e_G$. Pertanto $f = n$. □

Teorema 17. *Sia G un gruppo abeliano di ordine n, e sia $a \in G$ e un generatore di G. Per ogni $m \in \mathbb{N}$ si ha che $ord(a^{(m)}) = \frac{n}{\gcd(m,n)}$.*

Definizione 12. *Siano $(G, \oplus)$ e $(G', \oplus)$ due gruppi abeliani. Definiamo* omomorfismo *una funzione $\phi : G \to G'$ tale che $\phi(a \oplus b) = \phi(a) \oplus \phi(b)$ $\forall a, b \in G$.*

Due insiemi giocano un ruolo importante nella comprensione di un omomorfismo. Si tratta del *nucleo* e dell' *immagine* di ϕ. Chiamiamo *immagine* di ϕ l'insieme $Im(\phi) = \{\phi(a) : a \in G\}$ e *nucleo* di ϕ l'insieme $Ker(\phi) = \{a \in G : \phi(a) = e_{G'}\}$.

Teorema 18. *Siano* $(G, \oplus)$ *e* $(G', \oplus)$ *due gruppi abeliani e* $\phi : G \to G'$ *un omomorfismo allora* $Ker(\phi)$ *è un sottogruppo di* G*.*

Dimostrazione. Poichè sappiamo che $Ker(\phi) \subseteq G$, è sufficiente dimostrare che $Ker(\phi)$ è chiuso rispetto a $\oplus$ Siano $a, b \in Ker(\phi)$. Ma $\phi(a) = e_{G'}$ e $\phi(b) = e_{G'}$, quindi $\phi(a \oplus b) = \phi(a) \oplus \phi(b) = e_{G'} \oplus e_{G'} = e_{G'}$, da cui $a \oplus b \in Ker(\phi)$. □

Definizione 13. *Siano* $(G, \oplus)$ *e* $(G', \oplus)$ *due gruppi abeliani e* $\phi : G \to G'$ *un omomorfismo. Se* ϕ *è biettiva allora* ϕ *è detta* isomorfismo*.*

Definizione 14. *Si chiama* anello *un insieme non vuoto* A *con due operazioni algebriche binarie definite su di esso, una di addizione e una di moltiplicazione* $(A, +, \cdot)$ *tali che:*

1. $(A, +)$ *è un gruppo abeliano*
2. $(A, \cdot)$ *è un semigruppo, cioè* $a \cdot (b \cdot c) = (a \cdot b) \cdot c \ \forall a, b, c \in A$
3. *valgono le proprietà distributive della moltiplicazione rispetto all'addizione:*

$$a \cdot (b + c) = a \cdot b + a \cdot c \quad \forall a, b, c \in A$$

$$(b + c) \cdot a = b \cdot a + c \cdot a \quad \forall a, b, c \in A.$$

Definizione 15 (caratteristica di un anello)**.** *La* cararatteristica *di un anello* A *è il più piccolo positivo positivo intero* c*, tale che* $c \cdot 1_A = 0$*. Se* c *non esiste, la caratteristica di* A *è zero.*

Definizione 16. *Si definisce* corpo*, un anello* $(A, +, \cdot)$ *in cui gli elementi diversi da zero formano un gruppo rispetto alla moltiplicazione, tale grupo si dice* gruppo moltiplicativo *del corpo.*

Definizione 17 (campo)**.** *Si definisce* campo*, un corpo in cui la moltiplicazione è commutativa.*

Si può dimostrare che $(\mathbb{Q}, +, \cdot), (\mathbb{R}, +, \cdot), (\mathbb{C}, +, \cdot), (\mathbb{Z}_p, +, \cdot)$ con p numero primo, sono campi. Se $(\mathbb{K}, +, \cdot)$ è un campo, $\mathbb{K}' \subseteq \mathbb{K}$ e anche $(\mathbb{K}', +, \cdot)$ è un campo, allora $\mathbb{K}'$ è detto

sottocampo di $\mathbb{K}$. Un campo che ha un numero finito di elementi è detto *campo finito*. Un modo per descrivere un campo finito $\mathbb{K}$ consiste nel vederlo come ampliamento di un campo $\mathbb{F}$ contenuto in esso, ottenuto aggiungendovi una radice α di un polinomio irriducibile su $\mathbb{F}$. In tal caso, scriveremo che $\mathbb{K} = \mathbb{F}(\alpha)$ e si può dimostare che tale campo è isomorfo a $\mathbb{F}[x]/(h(x))$ (vedi paragrafo 1.4).
Inoltre, si dimostra che qualsiasi campo finito $\mathbb{F}$ ha cardinalità p^n, dove p è un numero primo ed n è un intero positivo. Generalmente un campo finito di p^n elementi è chiamato *campo di Galois* e si denota con $\mathbb{F}_{p^n}$ o $GF(p^n)$. Riportiamo di seguito, alcune proprietà utili sui campi finiti:

1. $\mathbb{F}_{p^m}$ è sottocampo di $\mathbb{F}_{p^n}$ se e solo se $m \mid n$
2. $\mathbb{F}_{p^n}$ ha caratteristica p
3. $\mid \mathbb{F}_{p^n}^* \mid = p^n - 1$.

1.3 Aritmetica Modulare

Da una analisi delle proprietà delle congruenze, si nota che l'operazione binaria "congruenza" è una relazione di equivalenza, perchè gode della proprietà riflessiva, simmetrica e transitiva. Data una relazione di equivalenza su un insieme A, il sottoinsieme che contiene tutti gli elementi equivalenti ad un elemento $a \in A$ è detto *classe di equivalenza* di a. Nel caso dei numeri interi, e definita la relazione $\cdot \equiv \cdot \pmod{n}$, denotiamo la classe di equivalenza contenente l'intero $a \in \mathbb{Z}$, con $[a]_n$. Storicamente, tale classe è chiamata classe dei residui modulo n. Quindi:

Definizione 18. $[a]_n = \{b \in \mathbb{Z} : b \equiv a \,(mod\, n)\}$.

Definiamo le operazioni di somma e prodotto sulle classi dei residui modulo n come segue:

Definizione 19. *definiamo con :*

$$[a]_n + [b]_n = [a + b]_n$$

$$[a]_n \cdot [b]_n = [a \cdot b]_n.$$

E' facile verificare che tali operazioni sono ben definite, cioè non dipendono dai rappresentati scelti delle classi. Usando la definizione dell'addizione modulo n possiamo costruire un gruppo, detto *gruppo additivo modulo* n, denotato $(\mathbb{Z}_n, +)$, dove $\mathbb{Z}_n = \{[a]_n : a \in \mathbb{Z}\}$. La cardinalità di questo gruppo è $\mid \mathbb{Z}_n \mid = n$.
Usando la definizione di moltiplicazione modulo n, possiamo costruire un gruppo, chiamato gruppo moltiplicativo modulo n, denotato $(\mathbb{Z}_n^*, \cdot)$ Dove $\mathbb{Z}_n^* = \{[a]_n \in \mathbb{Z}_n : \gcd(a, n) = 1\}$. La cardinalità di questo gruppo, come da definizione, è $\mid \mathbb{Z}_n^* \mid = \phi(n)$. Si può dimostrare che $(\mathbb{Z}_n, +)$ e $(\mathbb{Z}_n^*, \cdot)$ sono entrambi gruppi abeliani finiti e che $(\mathbb{Z}_n^*, \cdot)$ è ciclico se e solo se n è del tipo $2, 4, p^a, 2p^a$ per tutti i numeri primi dispari p e tutti gli interi positivi a.
Dopo aver definito le classi dei residui modulo n, concentriamo la nostra attenzione sulla risoluzione di alcune equazioni modulari. Cominciamo col trattare il problema della ricerca di tutte le x modulo n, che soddisfano la congruenza $ax \equiv b \, (mod\, n)$ assumendo che a e b siano dati.

Teorema 19. *L'equazione* $ax \equiv b \pmod{n}$ *ha soluzione se e solo se il* $\gcd(a, n) \mid b$.

Dimostrazione. Necessità : $\exists x_0 \in \mathbb{Z} : ax_0 \equiv b \pmod{n} \Rightarrow \gcd(a, n) \mid b$.
Dall'ipotesi abbiamo che $ax_0 - b = kn$ per qualche $k \in \mathbb{Z}$. dalla definizione di gcd possiamo scrivere che $a = \gcd(a, n)h$ e $n = \gcd(a, n)t$, per qualche $h, t \in \mathbb{Z}$. Da $ax_0 - b = kn$ possiamo ricavare b :

$$b = ax_0 - kn = \gcd(a, n)hx_0 - k\gcd(a, n)t = \gcd(a, n)(hx_0 - kt)$$

Quindi $\gcd(a, n) \mid b$.

Sufficienza : $\gcd(a, n) \mid b \Rightarrow \exists x_0 \in \mathbb{Z} : ax_0 \equiv b \pmod{n}$
Dall'ipotesi possiamo scrivere che $b = \gcd(a, n)k$ con $k \in \mathbb{Z}$. Ricordiamo che il massimo comune divisore tra due numeri si può sempre scrivere come loro combinazione lineare. Quindi: $\gcd(a, n) = aa' + nn'$ con $a', n' \in \mathbb{Z}$. Sostituendo abbiamo $b = aa'k + nn'k$ cioè $a(a'k) - b = (-n'k)n$. Ponendo $x_0 = n'k$ si ha la tesi. □

Teorema 20. *Se* $ax \equiv b \pmod{n}$ *è risolvibile e* x_0 *è una soluzione, allora l'equazione* $ax \equiv b \pmod{n}$ *ha esattamente* $\gcd(a,n)$ *soluzioni distinte modulo* n*, date da* $x_i = x_0 + i(n/\gcd(a,n))$ *per* $i = 0, ..., \gcd(a,n) - 1$.

Dimostrazione. Sia $d = \gcd(a,n)$ e sia x_0 una soluzione dell'equazione $ax \equiv b \pmod{n}$. Sappiamo che $d = aa' + nn'$ con $a', n' \in \mathbb{Z}$. Da cui si ha : $1 = (a/d)a' + (n/d)n'$. Quindi $\gcd(a/d, n/d) = 1$. Sia x_1 un' altra soluzione della congruenza. Allora $ax_1 - b = hn$ con $h \in \mathbb{Z}$, e $ax_0 - b = kn$ con $k \in \mathbb{Z}$. Sottraendo membro a membro, otteniamo $a(x_1 - x_0) = n(h-k)$ e, dividendo per d, ambo i membri, $(a/d)(x_1 - x_0) = (n/d)(h-k)$. Poichè $\gcd(a/d, n/d) = 1$, si deve avere che $(n/d) \mid (x_1 - x_0)$, cioè $x_1 - x_0 = j \cdot \frac{n}{d}$ per qualche $j \in \mathbb{Z}$. Adesso dimostriamo che le soluzioni distinte modulo n sono esattamente d.

Consideriamo due soluzioni congrue modulo n:

$$x_1 = x_0 + j\frac{n}{d}$$

$$x_2 = x_0 + j'\frac{n}{d}$$

con $j, j' \in \mathbb{Z}$. Allora $x_1 \equiv x_2 \pmod{n} \Leftrightarrow \frac{jn}{d} \equiv \frac{j'n}{d} \pmod{n} \Leftrightarrow j \equiv j' \pmod{d}$. Quindi per $0 \leq j < j' \leq d-1$, x_1 e x_2 sono distinte. □

Corollario 3. *L'equazione* $ax \equiv 1 \pmod{n}$ *ammette una soluzione se e solo se* $\gcd(a,n) = 1$.

Il valore che soddisfa l'equazione $ax \equiv 1 \pmod{n}$ è detto *inverso moltiplicativo di a modulo n* e si denota con $a^{-1} \mod n$. Quindi esiste l'inverso moltiplicativo di a modulo n se e solo se $\gcd(a,n) = 1$.

Teorema 21 (Teorema cinese del resto)**.** *Siano* $m_1, m_2, ..., m_r \in \mathbb{Z}$ *a due a due coprimi tra loro (cioè* $\gcd(m_i, m_j) = 1, \forall 1 \leq i < j \leq r$*) e siano* $a_1, a_2, ..., a_r \in \mathbb{Z}$*. Allora esiste uno e un solo intero* x *modulo* M*, con* $M = m_1 \cdot m_2 \cdots m_r$*, tale che :*

$$x \equiv a_i \pmod{m_i} \ \forall 1 \leq i \leq r.$$

Dimostrazione. Consideriamo la funzione $\theta : \mathbb{Z}_M \to \mathbb{Z}_{m_1} \times .... \times \mathbb{Z}_{m_r}$ definita come :

$$\theta(x) = (x \mod m_1, x \mod m_2, ..., x \mod m_r).$$

Per dimostrare il teorema, è sufficiente mostrare che θ è suriettiva e iniettiva.
Suriettività : Definiamo le quantità $M_i = M/m_i$ per ogni $1 \leq i \leq r$. Notiamo che $\gcd(M_i, m_i) = 1$, per ogni $1 \leq i \leq r$. Quindi esiste l'inverso moltiplicativo di M_i modulo m_i. Poniamo $y_i = M_i^{-1} \mod m_i$, per $1 \leq i \leq r$. Pertanto : $M_i y_i \equiv 1 \pmod{m_i}$, per $1 \leq i \leq r$. Verifichiamo adesso che la quantità

$$x = \sum_{i=1}^{r} a_i M_i y_i \mod M$$

soddisfa la relazione $\theta(x) = (a_1, a_2, ..., a_r)$. Adesso per $1 \leq i, j \leq r$, notiamo che :
se $i = j$, allora $a_i M_i y_i \equiv a_j \pmod{m_i}$
se $i \neq j$, allora $a_i M_i y_i \equiv 0 \pmod{m_j}$
Pertanto :

$$x = \sum_{i=1}^{r} a_i M_i y_i + kM \equiv \sum_{i=1}^{r} a_i M_i y_i \,(mod\, m_j) \equiv a_j \,(mod\, m_j)$$

cioè $a_j = x \mod m_j \; \forall 1 \leq j \leq r$.
Iniettività : Per dimostrare l'iniettività, notiamo che $\mid \mathbb{Z}_M \mid = \mid \mathbb{Z}_{m_1} \times .... \times \mathbb{Z}_{m_r} \mid = M$ e sia $\mathbb{Z}_M = \{x_1, x_2, ..., x_M\}$. Supponiamo per assurdo che $\exists x_i, x_j \in \mathbb{Z}_M : x_i \neq x_j$ e $\theta(x_i) = \theta(x_j)$. Allora, $\mid \mathbb{Z}_{m_1} \times .... \times \mathbb{Z}_{m_r} \mid = \mid \{\theta(x_1), ..., \theta(x_M)\} \mid < M$. Assurdo. □

Definizione 20 (funzione di Eulero)**.** *Sia $n \in \mathbb{N}^+$. Si definisce funzione di Eulero $\phi(n)$ il numero degli interi positivi che sono coprimi con n*

$$\phi(n) = \mid \{1 \leq a \leq n : \gcd(a, n) = 1\} \mid$$

Teorema 22. *Se $p \in \mathbb{P}$, allora $\phi(p) = p - 1$.*

Dimostrazione. Se p è primo allora tutti i numeri naturali da 1 a $p - 1$ sono primi con p. □

Teorema 23. *Se $p \in \mathbb{P}$ e a è un intero positivo, allora $\phi(p^a) = p^{a-1}(p - 1)$.*

Dimostrazione. I numeri tra 1 e p^a che non sono primi con p^a sono tutti e solo i multipli di p. Questi sono in totale $\frac{p^a}{p} = p^{a-1}$. Perciò il numero di numeri primi con p^a è $p^a - p^{a-1}$. Da cui segue immediatamente la tesi. □

Teorema 24. *Se $n = p_1^{a_1} \cdot p_2^{a_2} \cdots p_r^{a_r}$ è la fattorizzazione canonica di n, allora*

$$\phi(n) = \prod_{i=1}^{r} \phi(p_i^{a_i}) = n \prod_{i=1}^{r} (1 - 1/p_i).$$

Dimostrazione. Dimostriamo che dati due numeri interi positivi m, n con $\gcd(n, m) = 1$, si ha che

$$\phi(nm) = \phi(n) \cdot \phi(m).$$

Infatti, consideriamo la funzione

$$\theta : \mathbb{Z}_{nm} \to \mathbb{Z}_n \times \mathbb{Z}_m$$

$$[a]_{nm} \mapsto ([a]_n, [a]_m).$$

Tale funzione è ben definita, poichè $a \equiv a' \pmod{nm}$ implica $a \equiv a' \pmod{n}$ e $a \equiv a' \pmod{m}$ e dal Teorema cinese del resto, sappiamo che θ è una corrispondenza biunivoca. Inoltre, dal Lemma 2, $\gcd(a, nm) = 1$ se e solo se $\gcd(a, n) = 1$ e $\gcd(a, m) = 1$. Quindi θ è anche una corrispondenza biunivoca tra $\mathbb{Z}_{nm}^*$ e $\mathbb{Z}_n^* \times \mathbb{Z}_m^*$. Pertanto,

$$\phi(nm) = \mid \mathbb{Z}_{nm}^* \mid = \mid \mathbb{Z}_n^* \times \mathbb{Z}_m^* \mid = \phi(n) \cdot \phi(m).$$

Adesso, se $n = p_1^{a_1} \cdot p_2^{a_2} \cdots p_r^{a_r}$ è la fattorizzazione canonica di n, poichè $\gcd(p_i^{a_i}, p_j^{a_j}) = 1$, per ogni $1 \leq i \neq j \leq r$, concludiamo che :

$$\phi(n) = \prod_{i=1}^{r} \phi(p_i^{a_i}) = n \prod_{i=1}^{r} (1 - 1/p_i).$$

□

Definizione 21. *Siano n, k due numeri interi positivi, si definisce* ordine di n modulo k *e lo denotiamo con $ord_k(n)$, il più piccolo intero positivo t, se esiste, tale che $n^t \equiv 1 \pmod{k}$.*

Teorema 25 (Teorema del logaritmo discreto)**.** *Se g è un generatore di $\mathbb{Z}_n^*$ allora l'equazione $g^x \equiv g^y \pmod{n}$ vale se e solo se vale l'equazione $x \equiv y \pmod{\phi(n)}$.*

Dimostrazione. Necessità : $g^x \equiv g^y \pmod{n} \Rightarrow x \equiv y \pmod{\phi(n)}$
Poichè la sequenza di potenze di g genera ogni elementro di $[g]$ e $\mid [g] \mid = ord(g) = \phi(n)$, la sequenza di g è periodica con periodo $\phi(n)$. Quindi $y = x + k\phi(n)$ o $x = y + k\phi(n)$ con $k \in \mathbb{N}$. In ogni caso, segue la tesi.

Sufficienza : $x \equiv y \pmod{\phi(n)} \Rightarrow g^x \equiv g^y \pmod{n}$
Se $x \equiv y \pmod{\phi(n)}$ allora, $x = y + k\phi(n)$ per qualche $k \in \mathbb{Z}$. Quindi :

$$g^x \equiv g^{y+k\phi(n)} \,(mod\, n) \equiv g^y \cdot (g^{(\phi(n))})^k \,(mod\, n) \equiv g^y \cdot 1^k \,(mod\, n) \equiv g^y \,(mod\, n).$$

□

Teorema 26. *Sia $n = p_1^{a_1} \cdots p_r^{a_r}$ un numero intero dispari, e siano $y \in \mathbb{Z}$ tale che $\gcd(y, n) = 1$ e $k \in \mathbb{N}^+$. La congruenza, $x^k \equiv y \pmod{n}$ ha una soluzione se e solo se $\gcd(k, \phi(p_i^{a_i}))$ divide $ord_{g_i}(y)\ \forall i : 1 \leq i \leq r$ con g_i generatore di $\mathbb{Z}^*_{p_i^{a_i}}$. Se la congruenza ha una soluzione, allora ci sono esattamente*

$$\prod_{1 \leq i \leq r} \gcd(k, \phi(p_i^{a_i}))$$

soluzioni distinte modulo n.

Dimostrazione. La congruenza $x^k \equiv y \pmod{n}$ è equivalente a scrivere il sistema $x^k \equiv y \pmod{p_i^{a_i}}$ per ogni divisore primo p_i con $1 \leq i \leq r$. Dal Teorema del logaritmo discreto le congruenze possono essere scritte nella forma $k \cdot ord_{g_i}(x) \equiv ord_{g_i}(y) \pmod{\phi(p_i^{a_i})}$. Ognuna di queste congruenze ammette una soluzione se e solo se $d_i = \gcd(k, \phi(p_i^{a_i}))$ divide $ord_{g_i}(y)$ per ogni i.
Consideriamo una soluzione x_0 dell'equazione $x^k \equiv y \pmod{n}$. Poichè $\gcd(y, n) = 1$ allora esiste l'inverso moltiplicativo di x_0^k modulo n. Sia esso $(x_0^k)^{-1}$. A questo punto, per ogni altra soluzione x di $x^k(x_0^k)^{-1} = x^k(x_0^{-1})^k \equiv 1 \pmod{n}$ ovvero $(xx_0^{-1})^k \equiv 1 \pmod{p_i^{a_i}}$ per ogni $1 \leq i \leq r$ e dal teorema del logaritmo discreto:

$$k \cdot ord_{g_i}(xx_0^{-1}) \equiv 0 \pmod{\phi(p_i^{a_i})} \ \ \forall 1 \leq i \leq r$$

Notiamo che se $ord_{g_i}(xx_0^{-1}) = 0$ la conguenza sopra indicata è vera. Pertanto ciascuna di queste r equazioni ha esattamente d_i soluzioni

$$ord_{g_i}(xx_0^{-1}) = 0 + t_i\frac{\phi(p_i^{a_i})}{d_i} \text{ dove } t_i = 0, 1, ..., d_i - 1$$

Adesso, ogni r-pla di soluzioni per il sistema di equazioni $k \cdot ord_{g_i}(xx_0^{-1}) \equiv 0 \pmod{\phi(p_i^{a_i})}$ per ogni $1 \leq i \leq r$, rappresenta una soluzione dell'equazione $x^k \equiv y \pmod{n}$. Ma il numero di r-ple è $\prod_{i=1}^{r} d_i$. Segue la tesi. □

Corollario 4. *Se p è un numero primo dispari ed $a \geq 1$, allora l'equazione $x^2 \equiv 1 \pmod{p^a}$ ha solo due soluzioni, $x = 1$ e $x = -1$.*

Dimostrazione. Sappiamo che $\mathbb{Z}_{p^a}^*$ è ciclico ed ha generatore g. Per il lemma precendente, poichè $\gcd(2, \phi(p^a)) = 2$ e $2 \mid ord_g(1) = 0$, l'equazione $x^2 \equiv 1 \pmod{p^a}$ ammette esattamente 2 soluzioni. Per verifica diretta possiamo notare che le soluzioni sono 1 e -1. □

Definizione 22. *Un numero x è una radice quadrata non banale di 1 modulo n, se soddisfa l'equazione $x^2 \equiv 1 \pmod{n}$ con $x \neq 1$ e $x \neq -1$.*

Teorema 27. *Se esiste una radice quadrata non banale di 1 modulo n, allora n è composto.*

Dimostrazione. Per assurdo, se n fosse primo, per il corollario precedente, l'equazione $x^2 \equiv 1 \,(mod\, n)$ ammetterebbe due sole soluzioni $x = 1$ e $x = -1$ e quindi nessuna soluzione non banale. □

Lemma 3. *Se p è un numero primo e a un intero qualsiasi non divisibile per p, allora esiste un unico intero $b \in \{1, 2, ..., p-1\}$ tale che $ab \equiv 1 \pmod{p}$.*

Dimostrazione. Moltiplichiamo i numeri $1, 2, ..., p-1$, per a e consideriamo i resti $r_1, r_2, ..., r_{p-1}$ che si ottengono dividendo ciascun prodotto per p. Supponiamo per assurdo che $r_i = r_j$ per qualche $i > j$. Allora $ai \equiv aj \pmod{p}$ e quindi $a(i-j) \equiv 0 \pmod{p}$. Cioè p deve dividere $a(i-j)$. Ma p non può dividere a (per ipotesi), né $i-j$, perchè i e j sono interi

positivi minori di p e quindi $0 < i - j < p$. Quindi tutti i numeri r_i sono distinti, ed uno di questi è necessariamente uguale ad 1. □

Teorema 28 (Wilson)**.** *Un intero p è primo se e solo se* $(p-1)! \equiv -1 \pmod{p}$.

Dimostrazione. Necessità : Per $p = 2, 3$ il risultato è ovvio. Supponiamo dunque $p \geq 5$ e consideriamo il prodotto $(p-1)! = 1 \cdot 2 \cdots (p-1)$. Dal Lemma 3, possiamo associare a ciascun intero $2 \leq a \leq p-2$ il suo inverso moltiplicativo $b \in \{1, 2, ..., p-1\}$ che è unico e diverso da a. Infatti, i soli elementi di $\mathbb{Z}_p^*$ che sono uguali al proprio inverso moltiplicativo in $\mathbb{Z}_p^*$ sono 1 e $p-1$. Questo ci assicura che $(p-1)! \equiv 1 \cdot (p-1) \pmod{p}$ e quindi $(p-1)! \equiv -1 \pmod{p}$,

Sufficienza : Supponiamo per assurdo che p sia composto, cioè che esiste un intero $1 < d < p-1$ tale che $d \mid p$. Allora $d \mid (p-1)!$. Poichè, per ipotesi, $p \mid (p-1)!+1$, allora $d \mid (p-1)!+1$, ma $d \mid (p-1)!$ e pertanto $d = 1$. Assurdo. □

Teorema 29. *Siano $k, m \in \mathbb{Z}$ tali che* $\gcd(k, m) = 1$*. Se $\{a_1, a_2, ..., a_m\}$ è un insieme di residui incongruenti modulo m allora anche $\{ka_1, ka_2, ..., ka_m\}$ è un insieme di residui incongruenti modulo m.*

Dimostrazione. Per ipotesi sappiamo che $a_i \not\equiv a_j \pmod{m}$ $\forall i, j : 1 \leq i < j \leq m$. Se per assurdo, esistessero f, g con $1 \leq f < g \leq m$ tali che $ka_f \equiv ka_g \pmod{m}$, allora $m \mid k(a_f - a_g)$ e poichè $\gcd(m, k) = 1$ si avrebbe $a_f \equiv a_g \pmod{m}$, assurdo. □

Corollario 5. *Siano $k, m \in \mathbb{Z}$ tali che* $\gcd(k, m) = 1$*. Se $\mathbb{Z}_m^* = \{a_1, a_2, ..., a_{\phi(m)}\}$ è l'insieme completo dei residui primi modulo m, allora anche $\{ka_1, ka_2, ..., ka_{\phi(m)}\}$ è un insieme completo di residui primi modulo m.*

1.4 Cenni sulla teoria dei polinomi

Definizione 23 (funzione polinomiale)**.** *Sia K un qualunque campo, definiamo* funzione polinomiale *su K, una funzione $p : K \to K$, tale che $\exists n \in \mathbb{N}$ e degli elementi $a_i \in K$ tali che $p(\alpha) = a_0\alpha^0 + a_1\alpha^1 + ... + a_n\alpha^n$, per ogni $\alpha \in K$.*

Detto F l'insieme di tutte le funzioni polinomiali di K in se stesso, è possibile definire in tale insieme la somma e il prodotto, tenendo conto della definizione di somma e prodotto nel campo K:

$$(p+q)(x) = p(x) + q(x), \ \forall x \in K$$

$$(p \cdot q)(x) = p(x) \cdot q(x), \ \forall x \in K.$$

In tal modo F assume le caratteristiche di anello commutativo. Due funzioni polinomiali sono uguali se assumono gli stessi valori in corrispondenza di ogni $x \in K$. L'elemento neutro (lo *zero*) rispetto alla somma, è quella funzione polinomiale che associa ad ogni $x \in K$ l'elemento neutro di K. La funzione *unità* di F è la funzione polinomiale che associa ad ogni $x \in K$, l'elemento unità di K, ossia la funzione costante uguale ad 1.

Definizione 24 (polinomio)**.** *Un* polinomio $p(x)$ *a coefficienti nel campo K, è un'espressione del tipo :*

$$p(x) = a_0 + a_1x + a_2x^2 + ... + a_nx^n$$

con $a_i \in K$ e x un'indeterminata. L'insieme dei polinomi in K si indica con il simbolo $K[x]$.

I concetti di polinomio e funzione polinomiale, sono chiaramente diversi. Definita l'applicazione $\phi : K[x] \to F$ che associa ad ogni polinomio $p(x) = a_0 + a_1x + a_2x^2 + ... + a_nx^n \in K[x]$ la funzione polinomiale $p(x)$ che per ogni $c \in K$ manda in $p(c) = a_0 + a_1c + a_2c^2 + ... + a_nc^n \in K$, in generale è suriettiva ma non è iniettiva. Può accadere, infatti, che due polinomi diversi danno luogo alla stessa funzione polinomiale.

Definizione 25 (grado di un polinomio)**.** *Per* grado *di un polinomio $p(x) = a_0 + a_1x + a_2x^2 + ... + a_nx^n$, si intende l'intero n se $a_n \neq 0$.*

Il grado del polinomio $p(x)$ lo denotiamo con $deg(p)$ e a_n è detto *coefficiente direttivo.* Un polinomio che ha il coefficiente direttivo pari ad 1 è detto *monico.*
Le relazioni fra i gradi di due polinomi e quelli della loro somma e del loro prodotto sono le seguenti:

$$deg((p+q)) \leq \max(deg(p), deg(q)) \quad deg(p \cdot q) = deg(p) + deg(q) \text{ (purchè } p, q \neq 0).$$

Teorema 30. *Siano $f(x), g(x) \in K[x]$, con $g(x)$ monico. Allora esistono e sono unici due polinomi $q(x), r(x) \in K[x]$ tali che:*

$$f(x) = g(x)q(x) + r(x)$$

con $deg(r) < deg(g)$.

Dimostrazione.
Unicità : Supponiamo che $f(x) = g(x)q(x) + r(x) = g'(x)q(x) + r'(x)$, con $deg(r) < deg(g)$, $deg(r') < deg(g)$ e $q(x) \neq q'(x)$. Allora $r(x) - r'(x) = g(x)(q'(x) - q(x))$ e quindi:

$$deg(g) + deg(q' - q) = deg(g(q' - q)) = deg(r - r') \leq \max\{deg(r), deg(r')\} < deg(g)$$

e ciò è assurdo. Pertanto $q(x) = q'(x)$ e quindi $r(x) = r'(x)$.
Esistenza : Il caso $f(x) = 0$ è banale. Procediamo allora per induzione sul grado di $f(x)$. Se $deg(f) = 0$, allora $f(x) = g(x) \cdot 0 + f(x)$ è la divisione richiesta, purchè $deg(g) > 0$. Se anche $deg(g) = 0$, abbiamo $g(x) = 1$ e quindi $f(x) = 1 \cdot f(x) + 0$. Supponiamo $deg(f) = n > 0$ e che la tesi sia vera per i polinomi di grado minore di n. Indichiamo con m il grado di g.
Se $n < m$, allora $f(x) = g(x) \cdot 0 + f(x)$ è la divisione richiesta.
Se $n \geq m$, scriviamo $f(x) = a_0 + a_1x + ... + a_nx^n$ e consideriamo $h(x) = a_nx^{n-m}$. Allora il polinomio $f_1(x) = f(x) - g(x)h(x)$ ha grado minore di n : infatti $g(x)h(x)$ ha grado n ed ha lo stesso coefficiente direttivo di $f(x)$. Per l'ipotesi induttiva, possiamo scrivere $f_1(x) = g(x)q_1(x) + r(x)$, con $deg(r) < deg(g)$. Ma allora

$$f(x) = g(x)h(x) + f_1(x) = g(x)h(x) + g(x)q_1(x) + r(x) = g(x)(h(x) + q_1(x)) + r(x),$$

e si ha la tesi. □

Se $f(x) = g(x)h(x)$, si dice che $f(x)$ è divisibile per $g(x)$, oppure $g(x)$ divide $f(x)$ in $K[x]$. Si può altresi dimostrare il seguente teorema :

Teorema 31 (Esistenza e unicità del massimo comune divisore)**.** *Sia K un campo e siano $f(x), g(x) \in K[x]$ due polinomi non nulli. Allora esiste un unico polinomio monico $d(x)$ tale che :*

1. *$d(x)$ divide $f(x)$ e $g(x)$;*

2. *ogni divisore comune di $f(x)$ e $g(x)$ divide $d(x)$. Inoltre esistono due polinomi $h(x), k(x) \in K[x]$ tali che*

$$d(x) = h(x)f(x) + k(x)g(x) \text{ (identità di Bezout per i polinomi).}$$

Tale polinomio monico $d(x)$ si chiama massimo comune divisore *di $f(x)$ e $g(x)$ e si scrive $d(x) = \gcd(f(x), g(x))$.*

Definizione 26. *Se $f(x) \in K[x]$, un elemento $a \in K$ si dice che è una* radice o zero *di $f(x)$ se risulta $f(a) = 0$.*

Teorema 32 (Ruffini)**.** *Se $f(x) \in K[x]$ ed $a \in K$, a è una radice di $f(x) \Leftrightarrow (x-a) \mid f(x)$*

Dimostrazione.
Necessità : Dividiamo $f(x)$ per $(x-a)$, si ha : $f(x) = (x-a)q(x)+r(x)$, con $deg(r) = 0$. Essendo a radice di $f(x)$ otteniamo $0 = f(a) = 0q(a) + r(a) = r(a)$ e quindi $r(x) = 0$, da cui $(x - a) \mid f(x)$.
Sufficienza : Se $(x - a) \mid f(x)$ si ha $f(x) = (x - a)q(x)$. Sostituendo ad x il valore di a otteniamo $f(a) = 0$, cioè a è radice di $f(x)$. □

Corollario 6. *Un polinomio $f(x) \in K[x]$ di grado $n \geq 0$ ha al più n radici.*

Dimostrazione. Dimostriamo il teorema per induzione su n.
Caso base : se $n = 0$, allora $f(x)$ è costante e non ha radici.
Passo induttivo : se $n > 0$, allora $f(x)$ o non ha radici e il teorema è vero, oppure ha una radice a e, per il teorema di Ruffini, si ha $f(x) = (x - a)q(x)$ in cui $q(x)$ ha grado $n - 1$. Poichè le radici di $f(x)$ sono a e le radici di $q(x)$, per ipotesi induttiva, sono al più $n - 1$, segue che $f(x)$ ha al più n radici. □

Definizione 27 (Riducibilità)**.** *Sia $f(x) \in K[x]$ un polinomio tale che $deg(f) \geq 2$. Si dice che $f(x)$ è* riducibile *in $K[x]$ se esistono $g(x)$ e $h(x)$ non costanti (cioè di grado maggiore o uguale ad uno) tali che $f(x) = g(x) \cdot h(x)$.*

Definizione 28. *Si dice che* $f(x)$ *è* irriducibile *in* $K[x]$ *se non è costante e non esistono* $g(x)$ *e* $h(x) \in K[x]$ *con* $deg(g) \geq 1$ *e* $deg(h) \geq 1$ *tali che* $f(x) = g(x) \cdot h(x)$.

Definizione 29. *Sia* $p(x) \in K[x]$ *un polinomio di grado* n. *Diciamo che due polinomi* $f(x)$ *e* $g(x)$ *sono* congrui *modulo* $p(x)$ *in* $K[x]$ *e scriveremo*

$$f(x) \equiv g(x) \pmod{p(x)} \text{ in } K[x]$$

se esiste un opportuno polinomio $h(x) \in K[x]$ *tale che* $f(x) - g(x) = h(x) \cdot p(x)$.

In particolare useremo la notazione $f(x) \equiv g(x) \pmod{p(x), n}$ per indicare che $f(x) \equiv g(x) \pmod{p(x)}$ in $\mathbb{Z}_n[x]$, e la notazione $f(x) \equiv g(x) \pmod{n}$ per indicare che i coefficienti dei termini di ugual grado in $f(x)$ e $g(x)$ sono congrui modulo n.

Osserviamo che ogni polinomio è congruo $p(x)$ al resto della sua divisione per $p(x)$. Si verifica facilmente che la congruenza modulo $p(x)$ è una relazione di equivalenza in $K[x]$. Indicheremo la classe di equivalenza del polinomio $f(x)$ con $\overline{f(x)}$ e l'insieme quoziente di $K[x]$ rispetto a questa relazione con $K[x]/(p(x))$.
Possiamo a questo punto definire nell'insieme quoziente $K[x]/(p(x))$ due operazioni, di somma e prodotto, nel seguente modo :

$$\overline{f(x)} + \overline{g(x)} = \overline{f(x) + g(x)} \text{ e } \overline{f(x)} \cdot \overline{g(x)} = \overline{f(x) \cdot g(x)}.$$

Queste operazioni sono ben definite, cioè non dipendono dai rappresentanti scelti. Infatti, se

$$f_1(x) \equiv f_2(x) \pmod{p(x)} \text{ e } g_1(x) \equiv g_2(x) \pmod{p(x)},$$

allora

$$f_1(x) + g_1(x) \equiv f_2(x) + g_2(x) \pmod{p(x)}$$

$$f_1(x)g_1(x) \equiv f_2(x)g_2(x) \pmod{p(x)}.$$

Con queste operazioni, l'insieme quoziente $K[x]/(p(x))$ soddisfa la definzione di anello commutativo.

Teorema 33. *Sia $h(x)$ un polinomio irriducibile in $\mathbb{Z}_p[x]/(h(x))$ di grado d, allora l'anello $\mathbb{Z}_p[x]/(h(x))$ è un campo finito con p^d elementi.*

Dimostrazione. Ciascun polinomio in $\mathbb{Z}_p[x]/(h(x))$ ha grado minore di d e coefficienti in $\mathbb{Z}_p$. Pertanto in $\mathbb{Z}_p[x]/(h(x))$ ci sono esattamente p^d classi distinte di residui modulo $h(x)$ in $\mathbb{Z}_p[x]$. Adesso, dobbiamo provare che qualsiasi polinomio non identicamente nullo $f(x)$ in $\mathbb{Z}_p[x]/(h(x))$ ha un inverso moltiplicativo in $\mathbb{Z}_p[x]/(h(x))$. Poichè il grado di $f(x)$ è minore di d e $h(x)$ è irriducibile, noi abbiamo $gcd(f(x), h(x)) = 1$ e quindi dall'identità di Bezout segue che $f(x)$ ha inverso moltiplicativo in $\mathbb{Z}_p[x]/(h(x))$. □

Definizione 30 (radice r-esima dell'unità)**.** *Una* radice r-esima dell'unità *è una soluzione dell'equazione $z^r = 1$ in $\mathbb{C}$.*

Si può dimostrare che esistono r soluzioni distinte per $z^r = 1$ in $\mathbb{C}$, e sono esattamente

$$e^{2\pi i/r} = \cos\left(\frac{2\pi}{r}\right) + i\sin\left(\frac{2\pi}{r}\right)$$

$$e^{2\pi i 2/r} = \cos\left(\frac{2\pi}{r}\cdot 2\right) + i\sin\left(\frac{2\pi}{r}\cdot 2\right)$$

$$.....$$

$$e^{2\pi i r/r} = \cos\left(\frac{2\pi}{r}\cdot r\right) + i\sin\left(\frac{2\pi}{r}\cdot r\right) = 1.$$

Spesso si denota $e^{2\pi i/r}$ con ξ_r, così $\xi_r, \xi_r^2, ..., \xi_r^r$ sono tutte le radici distinte di $z^r = 1$.

Definizione 31 (radice r-esima primitiva dell'unità)**.** *Una radice r-esima dell'unità è detta* primitiva *se è della forma ξ_r^k con* $\gcd(k, r) = 1$.

Teorema 34. *Sia $G = \{\xi_r, \xi_r^2, ..., \xi_r^r\}$ l'insieme di tutte le radici r-esime distinte dell'unità. Allora,*

$$\mu \in G \text{ è una radice r-esima primitiva dell'unità se e solo se } \mu^m \neq 1 \forall m < r.$$

Dimostrazione.

Necessità : Sia $\mu = \xi_r^k$ una radice r-esima primtitiva dell'unità, per qualche $k = 1, 2, ...r$, tale che $\gcd(k, r) = 1$. Supponiamo per assurdo che $\exists m < r$ tale che $\mu^m = 1$.

Cioè $(\xi_r^k)^m = e^{2\pi i \frac{km}{r}} = 1$. Ma $e^{2\pi i \frac{km}{r}} = 1$ si verifica se e solo se $\cos\left(2\pi\frac{km}{r}\right) = 1$ e $\sin\left(2\pi\frac{km}{r}\right) = 0$ cioè se e solo se $\frac{km}{r} \in \mathbb{N}$. Ma poiché $\gcd(k,r) = 1$, segue che r divide m. Assurdo perchè $m < r$.

Sufficienza : Sia $\mu = \xi_r^k$ per qualche $k = 1, 2, ...r$ tale che $\mu^m \neq 1$ per ogni $m < r$. Supponiamo per assurdo che $\gcd(k,r) = d \neq 1$. Allora $(\xi_r^k)^{\frac{r}{d}} = (\xi_r^r)^{\frac{k}{d}} = 1$. Assurdo perchè abbiamo trovato un $m = \frac{r}{d} < r$ tale che $\mu^m = 1$. □

Definizione 32 (polinomio ciclotomico r-esimo)**.** *Sia r un intero positivo, definiamo* polinomio ciclotomico r-esimo *il polinomio monico*

$$Q_r(x) = \prod_{\substack{j=1 \\ \gcd(j,r)=1}}^{r} (x - \xi_r^j)$$

le cui radici sono esattamente le r-esime radici primitive dell'unità.

Teorema 35. *Sia r un intero positivo, e $Q_r(x)$ il polinomio ciclotomico r-esimo. Allora,*
(a) . $x^r - 1 = \prod_{d|r} Q_d(x)$
(b) . $Q_r(x)$ è monico e $Q_r(x) \in \mathbb{Z}[x]$.

Dimostrazione. (a) . $x^r - 1 = \prod_{d|r} Q_d(x)$. Consideriamo l'insieme $G = \{\xi_r, \xi_r^2, ..., \xi_r^r\}$ di tutte le radici r-esime distinte dell'unità e mostriamo che G contiene tutte le radici d-esime distinte dell'unità per ogni divisore d di r. Infatti, consideriamo i numeri $\mu_j = \xi_r^{\frac{r}{d}\cdot j}$ per ogni $j = 1, .., d$. Tali numeri appartengono a G, e quindi sono tutti distinti. Inoltre, $(\xi_r^{\frac{r}{d}j})^d = (\xi_r^r)^j = 1$. Chiaramente, dal Teorema 34, $\mu \in G$ è una radice primitiva d-esima e e solo se $ord(\mu) = d$, dove $ord(\mu)$ è il più piccolo intero positivo m tale che $\mu^m = 1$. Quindi per ogni divisore d di r,

$$Q_d(x) = \prod_{\substack{\mu \in G \\ ord(\mu)=d}} (x - \mu)$$

e infine :

$$x^r - 1 = \prod_{\mu \in G} (x - \mu) = \prod_{d|r} \left(\prod_{\substack{\mu \in G \\ ord(\mu)=d}} (x - \mu) \right) = \prod_{d|r} Q_d(x).$$

(b) . Dimostriamo questa parte del teorema per induzione su r.

Caso base : $Q_1(x) = x - 1 \in \mathbb{Z}[x]$.

Passo induttivo : Supponiamo vera la tesi per $k < r$ e sia

$$f(x) = \prod_{\substack{d|r \\ d \neq r}} Q_d(x).$$

Allora per ipotesi induttiva $f(x) \in \mathbb{Z}[x]$ e dalla (a), $x^r - 1 = f(x) \cdot Q_r(x)$. Ma $x^r - 1 \in Z[x]$ e $f(x)$ è monico. Di conseguenza, la divisione in $\mathbb{Z}[x]$ implica che $x^r - 1 = f(x)h(x) + r(x)$ per qualche $h(x), r(x) \in \mathbb{Z}[x]$. Per l'unicità del quoziente e del resto nella divisione in $\mathbb{Z}[x]$, si ha che $r(x) = 0$ e $Q_r(x) = h(x) \in \mathbb{Z}[x]$. □

Teorema 36. *Sia A un anello commutativo di caratteristica $c > 0$ e sia $\alpha \in A$ tale che $Q_r(\alpha) = 0$. Allora $r = c^e \cdot ord_r(\alpha)$ con $e \geq 0$.*

Dimostrazione. Poichè $Q_r(x)$ divide $x^r - 1$, abbiamo che $\alpha^r = 1$ e quindi $\alpha^r \equiv 1 \pmod{c}$. Pertanto $ord_c(\alpha)$ divide r e così possiamo scrivere che $r = c^e \cdot t \cdot ord_c(\alpha)$ per qualche intero positivo t non divisibile per c. Sia $s = c^e \cdot ord_c(\alpha)$ e assumiamo $t > 1$. Allora $\alpha^s = 1$ e osservando che

$$Q_r(x)g(x) = \frac{x^{st} - 1}{x^s - 1} = (x^s)^{t-1} + (x^s)^{t-2} + ... + (x^s)^2 + x^s + 1$$

per qualche $g(x) \in \mathbb{Z}[x]$, troviamo che

$$0 = Q_r(\alpha)g(\alpha) = (\alpha^s)^{t-1} + (\alpha^s)^{t-2} + ... + (\alpha^s)^2 + \alpha^s + 1 = t$$

cioè che c divide t. Assurdo. □

Teorema 37. *Sia G un sottogruppo finito del gruppo moltiplicativo di un campo K. Allora G è ciclico.*

Dimostrazione. Sia $m =\mid G \mid$. G è contenuto nell'insieme delle radici di $x^m - 1$ in K, il quale ha al più m elementi. Così otteniamo che $x^m - 1 = \prod_{\alpha \in G}(x - \alpha)$. Quindi, $Q_m(x)$ ha una radice $\beta \in G$ perchè $Q_m(x)$ divide $x^m - 1$. Se K ha caratteristica $p > 0$, allora p non è divisore di m perchè $x^m - 1$ non ha radici multiple, e così $m = ord_p(\beta)$, dal Teorema 36. Se K ha caratteristica zero, allora la tesi è immediata. □

Teorema 38. *Sia p un numero primo e q una potenza di p. Se p non è un divisore di r, allora $Q_r(x) \in \mathbb{F}_q[x]$ è il prodotto di polinomi irriducibili di grado $ord_r(q)$.*

Dimostrazione. Sia $h(x)$ un generico fattore irriducibile di $Q_r(x) \in \mathbb{F}_q[x]$ e sia ζ una radice di $h(x)$. Allora ζ è radice di $Q_r(x)$. Così dal Teorema 36 segue che $r = ord_p(\zeta)$ e dal Teorema 37 si ha $\zeta \in \mathbb{F}_{q^{ord_r(q)}}$. Poichè $\mathbb{F}_q(\zeta) = \mathbb{F}_{q^{deg(h)}}$ è sottocampo di $\mathbb{F}_{q^{ord_r(q)}}$, sappiamo che $deg(h)$ divide $ord_r(q)$. Inoltre, $\zeta \in \mathbb{F}_q(\zeta)^* = \mathbb{F}^*_{q^{deg(h)}}$ e quindi

$$\zeta^{q^{deg(h)}-1} = 1 = \zeta^0 \Rightarrow q^{deg(h)} = 1 \Rightarrow q^{deg(h)} \equiv 1 \pmod{r}$$

da cui $ord_r(q)$ divide $deg(h)$. Così abbiamo dimostrato che $deg(h) = ord_r(q)$. □

Capitolo 2

Teoria dei Numeri

2.1 Richiami di Analisi Matematica

Teorema 39. *Sia $f : [a, b] \to \mathbb{R}$ continua e monotona nell'intervallo $[a, b]$ dove $a, b \in \mathbb{Z}$, allora:*

$$\min(f(a), f(b)) \leq \sum_{i=a}^{b} f(i) - \int_a^b f(x)dx \leq \max(f(a), f(b))$$

Teorema 40. *Sia $[a, b] \subseteq \mathbb{R}$ e $x_0, x_1, ..., x_k \subseteq [a, b]$ con $a = x_0 \leq x_1 \leq x_2 \leq ... \leq x_k = b$. Siano $f_1, f_2, ..., f_k$ delle funzioni, rispettivamente continue negli intervalli $[x_i, x_{i+1}]$. Data $f(x)$ definita come segue : $f(x) = f_i(x)$ se $x \in [x_i, x_{i+1}]$. Si ha che f è integrabile in $[a, b]$ e che :*

$$\int_a^b f(x)dx = \sum_{i=1}^{k} \int_{x_{i-1}}^{x_i} f_i(x)dx$$

2.2 Teorema di Chebyshev

Definizione 33 (funzione di Chebyshev)**.**

$$\vartheta(x) = \sum_{p \leq x} \ln p \text{ con } p \text{ primi}$$

Teorema 41 (Chebyshev)**.** *$\forall x \in \mathbb{R}$ e $x \geq 1$ si ha che $\vartheta(x) < 2x \ln 2$*

Dimostrazione. Per dimostrare la tesi basta dimostrare che $\forall n \in \mathbb{N}$ e $n \geq 1$, $\vartheta(n) < 2n \ln 2$. Infatti se vale quest'ultima affermazione è immeditato notare che : $\vartheta(x) = \lfloor x \rfloor < 2\lfloor x \rfloor \ln 2 \leq 2x \ln 2$. Quindi concentriamo l'attenzione sulla tesi $\vartheta(n) < 2n \ln 2 \, \forall n \in N$ e $n \geq 1$. Dato un intero positivo m, consideriamo il coefficiente binomiale :

$$\binom{2m+1}{m}$$

Notiamo che tutti i primi p tali che $m + 1 < p \leq 2m + 1$ dividono M. Inoltre nell'espansione binomiale $(1 + 1)^{2m+1}$, M compare 2 volte. Una volta come $\binom{2m+1}{m}$ e una volta come $\binom{2m+1}{m+1}$. Quindi $M < \frac{2^{2m+1}}{2} = 2^{2m}$. Adesso:

$$\vartheta(2m + 1) - \vartheta(m + 1) = \sum_{m+1 < p \leq 2m+1} \ln p \leq \ln M < 2m \ln 2$$

Dimostriamo adesso il teorema per induzione su n.

Caso base : $\vartheta(1) = 0 < 2\ln 2$

Passo induttivo : Supponiamo vera la tesi $\forall q < n$ e dimostriamola per n.

Se n è pari : $\vartheta(n) = \vartheta(n-1) < 2(n-1)\ln 2 < 2n\ln 2$ Se n è dispari : sia $n = 2m+1$

$$\vartheta(n) = \vartheta(2m+1) = \vartheta(2m+1) - \vartheta(m+1) + \vartheta(m+1) < 2m\ln 2 + 2(m+1)\ln 2 =$$

$$= (2m + 2m + 2)\ln 2 = 2(2m+1)\ln 2 = 2n\ln 2$$

□

Definizione 34. *Dato n intero positivo e p un numero primo, definiamo $\nu_p(n)$ come la potenza alla quale un primo p divide n, cioè $\nu_p(n) = x \in \mathbb{N}$ se $p^x \mid n$ e $p^{x+1} \nmid n$*

Teorema 42. *Sia n un intero positivo. $\forall p \in \mathbb{P}$ si ha che :*

$$\nu_p(n!) = \sum_{k\geq 1} \lfloor \frac{n}{p^k} \rfloor$$

Dimostrazione. Osserviamo che la sequenza di numeri 1, 2, ...,n include esattamente $\lfloor \frac{n}{p} \rfloor$ mulipli di p, $\lfloor \frac{n}{p^2} \rfloor$ multipli di p^2 e così via. Definiamo dunque la sequenza di numeri:

$$\alpha_1 = \frac{n!}{p^{\lfloor \frac{n}{p} \rfloor}} \ , \ \alpha_2 = \frac{\alpha_1}{p^{\lfloor \frac{n}{p^2} \rfloor}} \ , \ ..... \ , \ \alpha_k = \frac{\alpha_{k-1}}{p^{\lfloor \frac{n}{p^k} \rfloor}} \text{ con } k = \lceil \log_p(n!) \rceil$$

Noi sappiamo che : $p \nmid \alpha_k$ e che $n! = \alpha_k \cdot p^{\nu_p(n!)}$. Quindi :

$$p^{\nu_p(n!)} = \frac{n!}{\alpha_k} = \frac{n!}{\left(\frac{\alpha_{k-1}}{p^{\lfloor \frac{n}{p^k} \rfloor}}\right)} = \frac{n!}{\frac{\left(\frac{\alpha_{k-1}}{p^{\lfloor \frac{n}{p^{k-1}} \rfloor}}\right)}{p^{\lfloor \frac{n}{p^k} \rfloor}}} = ... =$$

$$= \frac{n!}{\frac{n!}{p^{\sum_{i \leq \lceil \log_p(n!) \rceil} \lfloor \frac{n}{p^i} \rfloor}}} = p^{\sum_{i \leq \lceil \log_p(n!) \rceil} \lfloor \frac{n}{p^i} \rfloor}$$

Quindi $\nu_p(n!) = \sum_{i \leq \lceil \log_p(n!) \rceil} \lfloor \frac{n}{p^i} \rfloor$. Ma per $i > \lceil \log_p(n!) \rceil$ abbiamo che : $p^i > p^{\lceil log_p(n!) \rceil} \geq n! > n$. Ciò significa che $\lfloor \frac{n}{p^i} \rfloor = 0 \ \forall i > k$. Segue la tesi. □

2.3 Teorema di Mertens

Teorema 43 (Mertens)**.**

$$\sum_{p\leq x}\frac{\ln p}{p}=\ln x+\mathcal{O}(1) con\ p\ primi$$

Dimostrazione. Sia $n=\lfloor x\rfloor$ sappiamo dal Teorema di fattorizzazione, che $n!=\prod_{p\leq n}p^{\nu_p(n!)}$. Calcoliamo il $\ln(n!)$:

$$\ln(n!)=\ln\left(\prod_{p\leq n}p^{\nu_p(n!)}\right)=\sum_{p\leq n}\ln(p^{\nu_p(n!)})=\sum_{p\leq n}\nu_p(n!)\ln p$$

Dal Teorema 42 possiamo scrivere :

$$\ln(n!)=\sum_{p\leq n}\sum_{k\geq 1}\lfloor\frac{n}{p^k}\rfloor\ln p=\sum_{p\leq n}\lfloor\frac{n}{p}\rfloor\ln p+\sum_{k\geq 2}\sum_{p\leq n}\lfloor\frac{n}{p^k}\rfloor\ln p$$

Adesso concentriamoci sulla quantità $\sum_{k\geq 2}\sum_{p\leq n}\lfloor\frac{n}{p^k}\rfloor\ln p$. Questa è uguale a :

$$\sum_{p\leq n}\ln p\sum_{k\geq 2}\lfloor\frac{n}{p^k}\rfloor\leq n\sum_{p\leq n}\ln p\sum_{k\geq 2}p^{-k}=n\sum_{p\leq n}\ln p\frac{1}{p(p-1)}\leq n\sum_{k\geq 2}\frac{\ln k}{k(k-1)}=\mathcal{O}(n)$$

Dal momento che $\lfloor\frac{n}{p}\rfloor=\frac{n}{p}+\mathcal{O}(1)$ e applicando il teorema di Chebyshev :

$$\ln(n!)=\sum_{p\leq n}\frac{n}{p}\ln p+\mathcal{O}\left(\sum_{p\leq n}\ln p\right)+\mathcal{O}(n)=\sum_{p\leq n}\frac{n}{p}+\vartheta(n)+\mathcal{O}(n)=n\sum_{p\leq n}\frac{\ln p}{p}+\mathcal{O}(n)$$

Ma il $\ln(n!)$ si può scrivere usando il Teorema 37 così :

$$\ln(n!)=\ln\left(\prod_{k=1}^{n}k\right)=\sum_{k=1}^{n}\ln k=\int_1^n\ln t\,dt+\mathcal{O}(\ln n)=n\ln n-n+\mathcal{O}(\ln n)$$

A questo punto uguagliamo i due secondi membri per il calcolo del $\ln(n!)$:

$$n\sum_{p\leq n}\frac{\ln n}{p}+\mathcal{O}(n)\simeq n\ln n-n+\mathcal{O}(\ln n)$$

dividiamo per n e otteniamo :

$$\sum_{p\leq n}\frac{\ln p}{p}\simeq\ln n+\mathcal{O}\left(\frac{\ln n}{n}\right)-1\simeq\ln n+\mathcal{O}(1)$$

□

2.4 Identità di Abel

Teorema 44 (Abel)**.** *Sia $c_k, c_{k+1}, ...$ una sequenza di numeri e $C(t) = \sum_{k \le i \le t} c_i$. Sia f(t) derivabile in [k, x] allora :*

$$\sum_{k \le i \le x} c_i f(i) = C(x)f(x) - \int_k^x C(t)f'(t)\,dt$$

Dimostrazione. Sia $n = \lfloor x \rfloor$, possiamo scrivere la sommatoria come :

$\sum_{i=k}^{n} c_i f(i) = c_k f(k) + c_{k+1} f(k+1) + ... + c_n f(n) =$
$= C(k)f(k) + [C(k+1) - C(k)]f(k+1) + ... + [C(n) - C(n-1)]f(n) =$
$= C(k)[f(k) - f(k+1)] + ... + C(n-1)[f(n-1) - f(n)] + C(n)f(n) =$
$= C(k)[f(k) - f(k+1)] + ... + C(n-1)[f(n-1) - f(n)] + C(n)[f(n) - f(x)] + C(x)f(x)$
Osserviamo che $\forall i = k, ..., n-1$, si ha che $C(t) = C(i)\ \forall i \le t < i+1$ e quindi :

$$C(i)[f(i) - f(i+1)] = -\int_i^{i+1} C(t)f'(t)\,dt$$

così come :

$$C(n)[f(n) - f(x)] = -\int_n^x C(t)f'(t)\,dt$$

Ricapitolando, per il Teorema 38 :

$$\sum_{i=k}^{n} c_i f(i) = -\sum_{i=k}^{n-1} \int_i^{i+1} C(t)f'(t)\,dt - \int_n^x C(t)f'(t)\,dt + C(x)f(x) =$$

$$= C(x)f(x) - \int_k^n C(t)f'(t)\,dt - \int_n^x C(t)f'(t)\,dt = C(x)f(x) - \int_k^x C(t)f'(t)\,dt$$

□

Teorema 45.

$$\sum_{p \le x} \frac{1}{p} = \ln\ln x + \mathcal{O}(1) \textit{con p primo}$$

Dimostrazione. Applichiamo l'identità di Abel con la sequenza di numeri :

$$c_i = \begin{cases} \frac{\ln i}{i} & \text{se i è primo} \\ 0 & \text{altrimenti} \end{cases}$$

e scegliendo come funzione $f(t) = \frac{1}{\ln t}$ abbiamo che : $C(t) = \sum_{2 \leq i \leq t} c_i = \sum_{p \leq t} \frac{\ln p}{p}$. Per il teorema di Mertens $C(t) = \ln t + \mathcal{O}(1)$. Adesso, l'identità di Abel ci dice che :

$$\sum_{p \leq x} \frac{1}{p} = \frac{C(x)}{\ln(x)} + \int_s^x \frac{C(t)}{t(\ln t)^2}\, dt = \left(1 + \mathcal{O}\left(\frac{1}{\ln x}\right)\right) + \int_2^x \frac{\ln t + \mathcal{O}(1)}{t(\ln t)^2}\, dt =$$

$$= \left(1 + \mathcal{O}\left(\frac{1}{\ln x}\right)\right) + \int_2^x \frac{dt}{t \ln t} + \mathcal{O}\left(\int_2^x \frac{dt}{t(\ln t)^2}\right) =$$

$$= 1 + \mathcal{O}(1/\ln x) + (\ln\ln x - \ln\ln 2) + \mathcal{O}(1/\ln 2 - 1/\ln x) = \ln\ln x + \mathcal{O}(1)$$

□

Teorema 46. *Sia $n \in \mathbb{N}$ e $n \geq 2$, allora $\binom{2n+1}{n} > 2^{n+1}$.*

Dimostrazione.

$$\binom{2n+1}{n} = \frac{(2n+1)!}{n!(n+1)!} = \frac{(n+2)}{2} \cdots \frac{(n+n)}{n}(2n+1) > 2^{n-1} \cdot 2^2 = 2^{n+1}.$$

□

Teorema 47 (formula di Stifel)**.** *Siano $n, k \in \mathbb{N}$ e $1 \leq k \leq n$, allora*

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}.$$

Dimostrazione.

$$\binom{n-1}{k-1} + \binom{n-1}{k} = \frac{(n-1)!}{(k-1)!(n-k)!} + \frac{(n-1)!}{k!(n-k-1)!} =$$

$$\frac{(n-1)!}{(k-1)!(n-k-1)!}\left(\frac{1}{n-k} + \frac{1}{k}\right) = \frac{(n-1)!}{(k-1)!(n-k-1)!}\frac{n}{(n-k)k} = \frac{n!}{k!(n-k)!} = \binom{n}{k}.$$

□

Corollario 7. *Siano $n, k \in \mathbb{N}$ e $1 \leq k \leq m \leq n$ allora $\binom{n}{k} \geq \binom{m}{k}$*

Dimostrazione. Poniamo $p = n - m$. Dalla relazione di Stifel, otteniamo che :

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1} = \binom{n-2}{k} + \binom{n-2}{k-1} + \binom{n-1}{k-1} =$$

$$= .... = \binom{n-p}{k} + \sum_{i=1}^{p} \binom{n-i}{k-1} = \binom{m}{k} + \sum_{i=1}^{p} \binom{n-i}{k-1}.$$

Da cui segue $\binom{n}{k} \geq \binom{m}{k}$. □

Lemma 4. *Siano $n, k \in \mathbb{N}$ e $0 \leq k \leq 2n$. Allora $\binom{2n}{n} \geq \binom{2n}{k}$.*

Dimostrazione. Suddividiamo la dimostrazione per casi.

1. Per $k = 0$, la tesi è banalmente vera.

2. Per $0 \leq k < n$, si ha $n! > k!$ e $n < 2n - k \leq 2n$, da cui segue che $(2n-k)! > n!$. Pertanto:

$$\binom{2n}{n} = \frac{(2n)!}{n! \cdot n!} = \frac{(2n)!(n+1)\cdots(2n-k)}{k!(k+1)\cdots n \cdot n!(n+1)\cdots(2n-k)} =$$

$$= (n+1)\cdots(2n-k) \cdot \frac{(2n)!}{k! \cdot (2n-k)!} = (n+1)\cdots(2n-k) \cdot \binom{2n}{k} \geq \binom{2n}{k}.$$

3. Per $n < k \leq 2n$, si ha $0 \leq 2n - k < n$ e quindi dal punto 2, concludiamo con :

$$\binom{2n}{k} = \binom{2n}{2n-k} \leq \binom{2n}{n}.$$

□

Lemma 5 (Nair)**.** *Siano $m, n \in \mathbb{N} : 1 \leq m \leq n$ e $d_n = lcm(1, 2, ..., n)$ allora*

$$m\binom{n}{m} \Big| d_n$$

Dimostrazione. Calcoliamo l'integrale $I = I(m,n) = \int_0^1 x^{m-1}(1-x)^{n-m}dx$ seguendo due strategie risolutive diverse :

1. Integrazione per parti :

$$I = \frac{1}{m}[x^m(1-x)^{n-m}]_0^1 + \frac{n-m}{m}\int_0^1 x^m(1-x)^{n-m-1}dx = \frac{n-m}{m}\int_0^1 x^m(1-x)^{n-m-1}dx =$$

$$= \frac{n-m}{m}\left\{\frac{n-m-1}{m+1}\int_0^1 x^{m+1}(1-x)^{n-m-1}dx\right\} =$$

$$= \frac{1}{m}\cdot\frac{(n-m)(n-m-1)}{(m+1)}\int_0^1 x^{m+1}(1-x)^{n-m-1}dx =$$

...

...

si ripete l'integrazione per parti per altre $(n-m-2)$ volte

...

...

$$= \frac{1}{m} \cdot \frac{(n-m)(n-m-1)\cdots 3\cdot 2}{(m+1)(m+2)\cdots(n-1)} \cdot \int_0^1 x^{n-1}(1-x)^0 dx =$$

$$= \frac{1}{m} \cdot \frac{(n-m)(n-m-1)\cdots 3\cdot 2}{(m+1)(m+2)\cdots(n-1)} \cdot \frac{1}{n}[x^n]_0^1 =$$

$$= \frac{1}{m} \cdot \frac{(n-m)(n-m-1)\cdots 3\cdot 2\cdot 1}{(m+1)(m+2)\cdots(n-1)\cdot n} =$$

$$= \frac{1}{m} \cdot \frac{(n-m)!}{\frac{n!}{m!}} = \frac{1}{\left(m\binom{n}{m}\right)}.$$

2. Esplicitando $(1-x)^{n-m}$ come potenza di binomio :

$$I = \int_0^1 x^{m-1}\left(\sum_{r=0}^{n-m}(-1)^r\binom{n-m}{r}x^r\right)dx = \sum_{r=0}^{n-m}\left[(-1)^r\binom{n-m}{r}\int_0^1 x^{m+r-1}dx\right] =$$

$$\sum_{r=0}^{n-m}\left[(-1)^r\binom{n-m}{r}\cdot\frac{1}{m+r}[x^{m+r}]_0^1\right] = \sum_{r=0}^{n-m}\left[(-1)^r\binom{n-m}{r}\cdot\frac{1}{m+r}\right].$$

Quindi :

$$\frac{1}{\left(m\binom{n}{m}\right)} = \sum_{r=0}^{n-m}\left[(-1)^r\binom{n-m}{r}\cdot\frac{1}{m+r}\right].$$

Moltiplichiamo primo e secondo membro per d_n

$$\frac{d_n}{\left(m\binom{n}{m}\right)} = \sum_{r=0}^{n-m}\left[(-1)^r\binom{n-m}{r}\cdot\frac{d_n}{m+r}\right]$$

Notiamo che il secondo membro è un numero intero. Infatti $\binom{n-m}{r} \in \mathbb{N}$ e $r \leq n-m$ da cui segue $m+r \leq n$ e $(m+r) \mid d_n$ per ogni $0 \leq r \leq n-m$. Segue la tesi.

□

Teorema 48 (Nair)**.** *Sia $k \in \mathbb{N}$ e $k \geq 7$ allora $d_k = lcm(1,2,...,k) \geq 2^k$.*

Dimostrazione. Dal lemma precedente sappiamo che per ogni numero naturale $n \geq 1$, si ha

$$n\binom{2n}{n} \Big| d_{2n} \text{ e } (n+1)\binom{2n+1}{n+1} \Big| d_{2n+1}.$$

Poichè $d_{2n} \mid d_{2n+1}$ deduciamo che

$$n\binom{2n}{n} \Big| d_{2n+1}.$$

Inoltre,

$$(n+1)\binom{2n+1}{n+1} = (n+1)\frac{2n+1}{n+1}\binom{2n}{n} = (2n+1)\binom{2n}{n}.$$

Per quanto è stato detto e poichè n e $2n+1$ sono coprimi, segue che :

$$n(2n+1)\binom{2n}{n} \Big| d_{2n+1},$$

e quindi $d_{2n+1} \geq n(2n+1)\binom{2n}{n}$. Adesso, dal seguente risultato

$$4^n = (1+1)^{2n} = \sum_{k=0}^{2n} \binom{2n}{k} \leq \sum_{k=0}^{2n} \binom{2n}{n} = (2n+1)\binom{2n}{n}$$

otteniamo che :

$$d_{2n+1} \geq n \cdot 4^n.$$

Per $n \geq 2$ si ha $d_{2n+1} \geq n \cdot 4^n \geq 2 \cdot 4^n = 2^{2n+1}$ e questo dimostra la tesi per tutti i numeri k dispari maggiori o uguali a 5.

Per $n \geq 4$ si ha $d_{2n+2} \geq d_{2n+1} \geq n \cdot 4^n \geq 4 \cdot 4^n = 2^{2n+2}$ e questo dimostra la tesi per tutti i numeri k pari maggiori o uguali a 10.

Per concludere la dimostrazione, facendo una verifica diretta per k=6 e k=8 si trova

$$d_8 = 840 \geq 256 = 2^8 \text{ e } d_6 = 60 \ngeq 64 = 2^6.$$

□

2.5 Teoria dei residui quadratici

Definizione 35 (residuo quadratico)**.** *Sia $p \in \mathbb{P}$ e $a \in \mathbb{Z}$ tale che $p \nmid a$. Si dice che a è un* residuo quadratico *di p se l'equazione $x^2 \equiv a \pmod{p}$ ha almeno una soluzione, altrimenti a è detto* residuo non quadratico *di p.*

Definizione 36 (simbolo di Legendre)**.** *Dati $a \in \mathbb{Z}$ e $p \in \mathbb{P}$ definiamo il* simbolo di Legendre, $\left(\frac{a}{p}\right)$ *come segue:*
$\left(\frac{a}{p}\right) = +1$ *se* $p \nmid a$ *e* $x^2 \equiv a \pmod{p}$ *ha soluzioni (cioè a è un residuo quadratico di p)*
$\left(\frac{a}{p}\right) = -1$ *se* $p \nmid a$ *e* $x^2 \equiv a \pmod{p}$ *non ha soluzioni (cioè a è un residuo non quadratico di p)*
$\left(\frac{a}{p}\right) = 0$ *se* $p \mid a$.

Teorema 49. *Siano $p \in \mathbb{P}$ dispari e $a, b \in \mathbb{Z}$ allora:*
(a) . $a \equiv b \pmod{p} \Rightarrow \left(\frac{a}{p}\right) = \left(\frac{b}{p}\right)$
(b) . $\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \pmod{p}$ *(criterio di Eulero)*
(c) . $\left(\frac{-1}{p}\right) = (-1)^{\frac{p-1}{2}} = \begin{cases} +1 & \text{se } p \equiv 1 \pmod 4 \\ -1 & \text{se } p \equiv 3 \pmod 4 \end{cases}$
(d). $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$

Dimostrazione. (a) . Sia $a \equiv b \pmod{p}$.
Se $\left(\frac{a}{p}\right) = 0$, allora $a \equiv 0 \pmod{p}$ e quindi $b \equiv 0 \pmod{p}$, da cui $\left(\frac{b}{p}\right) = 0$.
Se $\left(\frac{a}{p}\right) = 1$, allora $p \nmid a$ e $x^2 \equiv a \equiv b \pmod{p}$ per qualche x, da cui $\left(\frac{b}{p}\right) = 1$. Infine, se $\left(\frac{a}{p}\right) = -1$, allora $p \nmid a$ e $x^2 \not\equiv a \equiv b \pmod{p}$ per ogni x e quindi $\left(\frac{b}{p}\right) = -1$.
(b) . Se $p \mid a$ allora $\left(\frac{a}{p}\right) = 0 \equiv a^{\frac{p-1}{2}} \pmod{p}$.
Se $p \nmid a$ consideriamo separatamente i due casi che possono presentarsi :

1. $\left(\frac{a}{p}\right) = 1$. In tal caso esistono esattamente due soluzioni di $x^2 \equiv a \pmod{p}$. Infatti, sia c soluzione della congruenza $x^2 \equiv a \pmod{p}$, allora anche $(p-c) \equiv -c \pmod{p}$ è soluzione di $x^2 \equiv a \pmod{p}$. Inoltre l'insieme $\mathbb{Z}_p^* \backslash \{c, p-c\}$ può essere suddiviso in $(p-3)/2$ coppie distinte di interi distinti $(x_1, x_1') = (x_1, ax_1^{-1})$ tali che $x_1 x_1' \equiv a \pmod{p}$. Segue che :

$$(p-1)! \equiv c \cdot (p-c) \cdot a^{\frac{p-3}{2}} \equiv -c^2 a^{\frac{p-1}{3}} \equiv -a \cdot a^{\frac{p-1}{3}} \equiv -a^{\frac{p-1}{2}} \pmod{p}$$

   e dal Teorema di Wilson, $-a^{\frac{p-1}{2}} \equiv (p-1)! \equiv -1 \pmod{p}$, segue la tesi.

2. $\left(\frac{a}{p}\right) = -1$. In tal caso esistono $(p-1)/2$ coppie distinte di interi distinti (x_1, x_1')

con $x_1, x_1' \in \mathbb{Z}_p^*$ tali che $x_1 x_1' \equiv a \pmod{p}$. Quindi $a^{\frac{p-1}{2}} \equiv (p-1)! \equiv -1 \pmod{p}$ e segue la tesi.

(c) . Applicando la (b) con $a = -1$ otteniamo $\left(\frac{-1}{p}\right) \equiv (-1)^{(p-1)/2} \pmod{p}$
Se $(p-1)/2$ è pari allora $\left(\frac{-1}{p}\right) = 1$, ma $(p-1)/2$ è pari $\Leftrightarrow (p-1)/2 = 2k$ con k intero $\Leftrightarrow$ $(p-1) = 4k \Leftrightarrow p - 1 \equiv 0 \pmod{4} \Leftrightarrow p \equiv 1 \pmod{4}$.
Se $(p-1)/2$ è dispari allora $\left(\frac{-1}{p}\right) = -1$, ma $(p-1)/2$ è dispari $\Leftrightarrow (p-1)/2 = 2k +$ 1 con k intero $\Leftrightarrow (p-1) = 4k + 2 \Leftrightarrow p - 1 \equiv 2 \pmod{4} \Leftrightarrow p \equiv 3 \pmod{4}$.

(d) . Applicando la (b) ottendiamo :
$\left(\frac{ab}{p}\right) \equiv (ab)^{(p-1)/2} \equiv a^{(p-1)/2} b^{(p-1)/2} \equiv \left(\frac{a}{p}\right)\left(\frac{b}{p}\right) \pmod{p}$ e poichè $-2 \leq \left(\frac{ab}{p}\right) - \left(\frac{a}{p}\right)\left(\frac{b}{p}\right) \leq 2$ e $p > 2$ si ha che $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$. □

Corollario 8. *Sia $p \in \mathbb{P}$, per ogni $a \in \mathbb{Z}$*
Se $p \equiv 1 \pmod{4}$ allora $\left(\frac{-a}{p}\right) = \left(\frac{a}{p}\right)$.
Se $p \equiv 3 \pmod{4}$ allora $\left(\frac{-a}{p}\right) = -\left(\frac{a}{p}\right)$.

Dimostrazione. Se $p \equiv 1 \pmod{4}$ allora $(p-1)/2$ è pari e quindi dal criterio di Eulero $(-a)^{(p-1)/2} = (a)^{(p-1)/2}$ da cui segue la tesi.
Se $p \equiv 3 \pmod{4}$ allora $(p-1)/2$ è dispari e quindi dal criterio di Eulero $(-a)^{(p-1)/2} =$ $-(a)^{(p-1)/2}$ da cui segue la tesi. □

Lemma 6 (Lemma di Gauss)**.** *Sia p un primo dispari ed a un intero non divisibile per p. Consideriamo il sistema completo dei residui minimi in valore assoluto (modulo p) :*

$$\Sigma = \{-\frac{p-1}{2}, ..., -1, 0, 1, ..., \frac{p-1}{2}\}$$

e l'insieme :

$$S(a) = \{ka : 1 \leq k \leq \frac{p-1}{2}\}.$$

Indicato con s il numero di elementi di $S(a)$ congruenti (modulo p) agli interi negativi di Σ, si ha :

$$\left(\frac{a}{p}\right) = (-1)^s.$$

Dimostrazione. Scelti $1 \leq h, k \leq \frac{p-1}{2}$ notiamo che :

1. se $ka \equiv ha \pmod{p}$, dato che $p \nmid a$, allora $p \mid (h-k)$, da cui essendo $-\frac{p-1}{2} \leq h-k \leq \frac{p-1}{2}$ segue che $h = k$.

2. Se per assurdo $ka \equiv -ha \pmod{p}$, dato che $p \nmid a$, allora $p \mid (h+k)$ da cui essendo $0 < h+k < p$ segue che $k = -h$. Assurdo perchè $kh > 0$.

3. Non può risultare $ka \equiv 0 \pmod{p}$ dato che $p \nmid a$ e $p \nmid k$.

Quindi, per ogni k tale che $1 \leq k \leq \frac{p-1}{2}$ esiste un unico $r_k \in \Sigma$ tale che $r_k \equiv ka \pmod{p}$, e per quanto osservato sopra, l'insieme $\{r_1, ..., r_{\frac{p-1}{2}}\}$ è costituito da interi a due a due differenti in valore assoluto. Ne segue che gli insiemi $\{1, 2, ..., \frac{p-1}{2}\}$ e $\{\mid r_1 \mid, ..., \mid r_{\frac{p-1}{2}} \mid\}$ coincidono, e in base alla definizione di s, si ha:

$$\prod_{i=1}^{\frac{p-1}{2}} r_i = (-1)^s \prod_{i=1}^{\frac{p-1}{2}} \mid r_i \mid = (-1)^s \left(\frac{p-1}{2}\right)!.$$

Inoltre, usando il criterio di Eulero, otteniamo :

$$\prod_{i=1}^{\frac{p-1}{2}} r_i \equiv \prod_{i=1}^{\frac{p-1}{2}} ia \equiv a^{\frac{p-1}{2}} \left(\frac{p-1}{2}\right)! \equiv \left(\frac{a}{p}\right) \left(\frac{p-1}{2}\right)! \pmod{p}.$$

Per la transitività dell' uguaglianza, possiamo scrivere: $(-1)^s \left(\frac{p-1}{2}\right)! \equiv \left(\frac{a}{p}\right) \left(\frac{p-1}{2}\right)! \pmod{p}$ e poichè $p \nmid \left(\frac{p-1}{2}\right)!$ deve verificarsi che :

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

Da cui, essendo $p > 2$, segue la tesi. □

Corollario 9. *Sia p un primo dispari allora :*

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

Dimostrazione. Applichiamo il lemma di Gauss e contiamo il numero s di elementi dell'insieme $S(2) = \{2k : 1 \leq k \leq \frac{p-1}{2}\} = \{2, 4, ..., p-1\}$ congruenti modulo p agli interi negativi dell'insieme $\Sigma = \{-\frac{p-1}{2}, ..., -1, 0, 1, ..., \frac{p-1}{2}\}$. Notiamo che per ogni $\alpha \in S(2)$ tale che $\alpha < \frac{p-1}{2}$ si ha che $0 \leq \alpha \bmod p \leq \frac{p-1}{2}$. Mentre per tutti gli $\alpha \in S(2)$ tali

che $\alpha > \frac{p-1}{2}$ esiste un unico residuo negativo in Σ. Quindi la condizione necessaria e sufficiente affinchè esiste un residuo negativo modulo p di $\alpha \in S(2)$ con $\alpha = 2k$ è che $k > \frac{p-1}{4}$. Quindi possiamo determinare s, considerando 2 casi :

1. $\frac{p-1}{4}$ è un intero. In tal caso $s = \frac{p-1}{2} - \frac{p-1}{4} = \frac{p-1}{4}$. Se $\frac{p-1}{4}$ è dispari, cioè $p \equiv -3 \pmod 8$ otteniamo $\left(\frac{2}{p}\right) = -1$. Se $\frac{p-1}{4}$ è pari, cioè $p \equiv 1 \pmod 8$ otteniamo $\left(\frac{2}{p}\right) = 1$.

2. $\frac{p-1}{4}$ non è un intero, allora il più grande numero in $S(2)$ che ha un residuo positivo modulo p è $\frac{p-3}{2}$. Da cui otteniamo $s = \frac{p-1}{2} - \frac{p-3}{4} = \frac{p+1}{4}$. Se $\frac{p+1}{4}$ è dispari, cioè $p \equiv 3 \pmod 8$ otteniamo $\left(\frac{2}{p}\right) = -1$. Se $\frac{p+1}{4}$ è pari, cioè $p \equiv -1 \pmod 8$ otteniamo $\left(\frac{2}{p}\right) = 1$.

Infine, poichè $\frac{p^2-1}{8} = \frac{p+1}{2}\frac{p-1}{4}$ ha la stessa parità di s, segue la tesi. □

Corollario 10. *Siano p un primo dispari ed a un intero anch'esso dispari tale che* $\gcd(a,p) = 1$. *Allora*

$$\left(\frac{a}{p}\right) = (-1)^{\sigma_{a,p}} \text{ con } \sigma_{a,p} = \sum_{k=1}^{\frac{p-1}{2}} \left\lfloor \frac{ka}{p} \right\rfloor.$$

Dimostrazione. Come nel Lemma di Gauss, sia $S(a) = \{ka : 1 \leq k \leq \frac{p-1}{2}\}$. Dividendo gli elementi di $S(a)$ per p, si ottiene :

$$ka = q_k p + t_k \text{ con } q_k, t_k \in \mathbb{N} \text{ e } \leq t_k \leq p-1.$$

Segue che $\frac{ka}{p} = q_k + \frac{t_k}{p}$ e quindi $\left\lfloor \frac{ka}{p} \right\rfloor = q_k$. Pertanto si ha :

$$ka = \left\lfloor \frac{ka}{p} \right\rfloor \cdot p + t_k, \text{ con } 1 \leq k \leq \frac{p-1}{2}.$$

Denotiamo con $\{v_1, ..., v_\mu\}$ l'insieme $\{t_k : 1 \leq t_k \leq \frac{p-1}{2}$, al variare di k e con $1 \leq k \leq \frac{p-1}{2}\}$ e con $\{r_1, ..., r_s\}$ l'insieme $\{t_k : \frac{p+1}{2} \leq t_k \leq p-1$, al variare di k e con $1 \leq k \leq \frac{p-1}{2}\}$. Notiamo che s è lo stesso intero definito come nel Lemma di Gauss. Verifichiamo che l'insieme $\{v_1, ..., v_\mu, p-r_1, ..., p-r_s\}$ coincide con l'insieme $\{1, 2, ...\frac{p-1}{2}\}$. A tal scopo è sufficiente provare che $v_{i'} \not\equiv p - r_{j'} \pmod p$ con $1 \leq i' \leq \mu$ e $1 \leq j' \leq s$. Infatti, se

$v_{i'} \equiv ia \pmod p$ e $r_{j'} \equiv ja \pmod p$ dove $1 \leq i \neq j \leq \frac{p-1}{2}$, allora $(i+j)a \equiv v_{i'} + r_{j'} \pmod p$. Se per assurdo fosse $v_{i'} \equiv p - r_{j'} \pmod p$, allora $(i+j)a \equiv 0 \pmod p$ e dunque $i + j \equiv 0 \pmod p$, il che è assurdo.

Quindi :

$$\sum_{k=1}^{\frac{p-1}{2}} k = \sum_{i=1}^{\mu} v_i + \sum_{j=1}^{s} (p - r_j) = ps + \sum_{i=1}^{\mu} v_i - \sum_{j=1}^{s} r_j$$

ed anche:

$$\sum_{k=1}^{\frac{p-1}{2}} ka = \sum_{k=1}^{\frac{p-1}{2}} \left\lfloor \frac{ka}{p} \right\rfloor \cdot p + \sum_{i=1}^{\mu} v_i + \sum_{j=1}^{s} r_j.$$

Sottraendo la prima uguaglianza alla seconda, si ottiene:

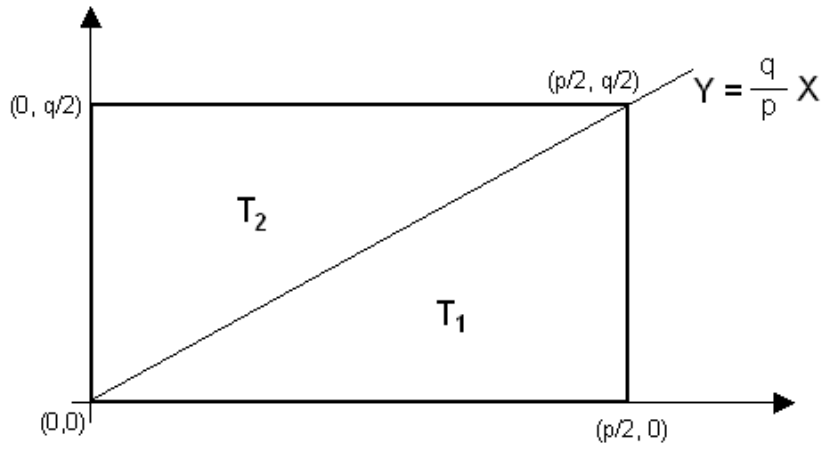
$$(a-1) \sum_{k=1}^{\frac{p-1}{2}} k = p \cdot (\sigma_{a,p} - s) + 2 \sum_{j=1}^{s} r_j.$$

Tenendo presente che $p \equiv a \equiv 1 \pmod 2$, si ha $0 \equiv \sigma_{a,p} - s \pmod 2$ e applicando il Lemma di Gauss si ha la tesi. □

Teorema 50 (Teorema di reciprocità quadratica)**.** *Siano p e q primi dispari distinti, allora:*

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

Dimostrazione. Costruiamo nel piano cartesiano un rettangolo $R = [0,0] \times [p/2, q/2]$.



L'idea della dimostrazione consiste nel contare, in due modi distinti, i punti a coordinate intere giacenti in R. Chiaramente un punto a coordinate intere (m, n) appartiene ad R se, e solo se, $1 \leq m \leq \frac{p-1}{2}$ e $1 \leq n \leq \frac{q-1}{2}$ ed essendo $\frac{p-1}{2} = \lfloor \frac{p}{2} \rfloor$ e $\frac{q-1}{2} = \lfloor \frac{q}{2} \rfloor$, un primo conteggio ci dice che tali punti sono $\frac{p-1}{2} \cdot \frac{q-1}{2}$.

Procediamo adesso al calcolo degli stessi punti con un altro metodo. Introduciamo l'equazione della diagonale del rettangolo $Y = \frac{q}{p}X$ e osserviamo che nessun punto di R a coordinate intere (m, n) giace sulla diagonale. In caso contrario, risulterebbe $n = \frac{q}{p}m$, dunque $pn = qm$ e pertanto $p \mid m$ e $q \mid n$. Ciò sarebbe in contrasto con il fatto che $1 \leq m \leq \frac{p-1}{2}$ e $1 \leq n \leq \frac{q-1}{2}$.

Se denotiamo allora con T_1 il sottoinsieme triangolare di R a coordinate intere al di sotto della diagonale, e con T_2 rispettivamente quello al di sopra della diagonale, è evidente che i punti cercati sono la somma di quelli giacenti in T_1 con quelli giacenti in T_2. Ora, se k è un intero tale che $1 \leq k \leq \frac{p-1}{2}$, il numero degli interi y tali che $0 < y < \frac{qk}{p}$ è dato da $\lfloor qk/p \rfloor$ e pertanto i punti di T_1 a coordinate intere e con ascissa k sono esattamente $\lfloor qk/p \rfloor$. Ne segue che i punti a coordinate intere in T_1 sono :

$$\sum_{k=1}^{\frac{p-1}{2}} \left\lfloor \frac{qk}{p} \right\rfloor .$$

Analogamente, i punti a coordinate intere in T_2 sono :

$$\sum_{k=1}^{\frac{q-1}{2}} \left\lfloor \frac{pk}{q} \right\rfloor .$$

In definitiva, abbiamo

$$\sum_{k=1}^{\frac{p-1}{2}} \left\lfloor \frac{qk}{p} \right\rfloor + \sum_{k=1}^{\frac{q-1}{2}} \left\lfloor \frac{pk}{q} \right\rfloor = \frac{p-1}{2} \cdot \frac{q-1}{2}.$$

Applicando il Corollario 10, abbiamo:

$$\left(\frac{p}{q}\right)\left(\frac{q}{p}\right) = (-1)^{\sum_{k=1}^{\frac{p-1}{2}} \left\lfloor \frac{qk}{p} \right\rfloor} \cdot (-1)^{\sum_{k=1}^{\frac{q-1}{2}} \left\lfloor \frac{pk}{q} \right\rfloor} = (-1)^{\frac{p-1}{2} \cdot \frac{q-1}{2}}.$$

□

Definizione 37 (simbolo di Jacobi)**.** *Dati $a, n \in \mathbb{Z}$, con n dispari e fattorizzazione $n = \pm p_1^{e_1} p_2^{e_2} \cdots p_s^{e_s}$, si definisce* simbolo di Jacobi $\left(\frac{a}{n}\right)$ *il numero*

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right)^{e_1} \left(\frac{a}{p_2}\right)^{e_2} \cdots \left(\frac{a}{p_s}\right)^{e_s}.$$

Teorema 51. *Siano $a, n \in \mathbb{Z}$, con n dispari avente fattorizzazione $n = \pm p_1^{e_1} p_2^{e_2} \cdots p_s^{e_s}$. Allora:*

(a). $\left(\frac{a}{n}\right) = 0$ *se e solo se* $\gcd(a, n) > 1$

(b). $\left(\frac{a}{n}\right) = \left(\frac{a}{|n|}\right)$

(c). se $a \equiv b \pmod{n}$, *allora* $\left(\frac{a}{n}\right) = \left(\frac{b}{n}\right)$

(d). $\left(\frac{ab}{n}\right) = \left(\frac{a}{n}\right)\left(\frac{b}{n}\right)$

(e). Se $\gcd(a, n) = 1$ *allora* $\left(\frac{a^2}{n}\right) = 1$.

Dimostrazione. (a). Se $\gcd(a, n) > 1$ allora $\exists p_j$ con $1 \leq j \leq s$ tale che $p_j \mid a$ e quindi $\left(\frac{a}{p_j}\right) = 0$, da cui $\left(\frac{a}{n}\right) = 0$. Viceversa se $\left(\frac{a}{n}\right) = 0$ allora $\exists p_j$ con $1 \leq j \leq s$ tale che $\left(\frac{a}{p_j}\right) = 0$. Cioè $p_j \mid a$ e quindi $\gcd(a, n) > 1$.

(b). Segue immediatamente dalla definizione del simbolo di Jacobi.

(c). Se $a \equiv b \pmod{n} \Rightarrow \forall 1 \leq j \leq s$ si ha che $a \equiv b \pmod{p_j} \Rightarrow \left(\frac{a}{p_j}\right) = \left(\frac{b}{p_j}\right) \forall 1 \leq j \leq s \Rightarrow \left(\frac{a}{n}\right) = \left(\frac{b}{n}\right)$.

(d) . $\left(\frac{ab}{n}\right) = \left(\frac{ab}{p_1}\right)^{e_1} \cdots \left(\frac{ab}{p_s}\right)^{e_s} = \left(\left(\frac{a}{p_1}\right)\left(\frac{b}{p_1}\right)\right)^{e_1} \cdots \left(\left(\frac{a}{p_s}\right)\left(\frac{b}{p_s}\right)\right)^{e_s} = \left(\frac{a}{p_1}\right)^{e_1} \cdots \left(\frac{a}{p_s}\right)^{e_s} \left(\frac{b}{p_1}\right)^{e_1} \cdots \left(\frac{b}{p_s}\right)^{e_s} = \left(\frac{a}{n}\right)\left(\frac{b}{n}\right)$.

(e). Se $\gcd(a, n) = 1 \Rightarrow \gcd(a^2, n) = 1 \Rightarrow \left(\frac{a^2}{n}\right) \neq 0$. Ma a è soluzione di tutte le equazioni $x^2 \equiv a^2 \pmod{p_j} \forall 1 \leq j \leq s$. Quindi tutti i simboli di legendre $\left(\frac{a}{p_j}\right)$ saranno uguali ad 1. □

Teorema 52. *Siano m, n interi positivi dispari, con fattorizzazione $n = p_1^{e_1} p_2^{e_2} \cdots p_s^{e_s}$ e $m = q_1^{f_1} q_2^{f_2} \cdots q_r^{f_r}$ allora:*

(a). $\left(\frac{-1}{n}\right) = (-1)^{\frac{n-1}{2}}$

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

(c). $\left(\frac{m}{n}\right) = \left(\frac{n}{m}\right)(-1)^{\frac{m-1}{2}\frac{n-1}{2}}$ *(teorema di reciprocità quadratica generalizzata)*

Dimostrazione. (a). $\left(\frac{-1}{n}\right) = \prod_{i=1}^{s} \left(\frac{-1}{p_i}\right)^{e_i} = (-1)^{\sum_{i=1}^{s} e_i \frac{p_i-1}{2}}$
Osserviamo che :

1. Poichè n è dispari può essere il prodotto di numeri primi congrui ad 1 modulo 4 o congrui a 3 modulo 4.

2. $\frac{n-1}{2}$ è pari $\Leftrightarrow n \equiv 1 \pmod 4 \Leftrightarrow$ è pari il numero di primi p_i congrui 3 modulo 4 (cioè tali che $\frac{p_i-1}{2}$ sono dispari) con e_i dispari.

3. $\frac{n-1}{2}$ è dispari $\Leftrightarrow n \equiv 3 \pmod 4$.

4. Il numero di primi p_i congrui 3 modulo 4 (cioè tali che $\frac{p_i-1}{2}$ sono dispari) con e_i dispari è pari, se e solo se $\sum_{i=1}^{s} e_i \frac{p_i-1}{2}$ è pari.

Pertanto $\sum_{i=1}^{s} e_i \frac{p_i-1}{2}$ e $\frac{n-1}{2}$ hanno la stessa parità, e quindi si ha la tesi.

(b). $\left(\frac{2}{n}\right) = \prod_{i=1}^{s} \left(\frac{2}{p_i}\right)^{e_i} = (-1)^{\sum_{i=1}^{s} e_i \frac{p_i^2-1}{8}}$
Osserviamo che :

1. Poichè n è dispari può essere il prodotto di numeri primi congrui ad ± 1 modulo 8 o congrui a ± 3 modulo 8.

2. $\frac{n^2-1}{8}$ è dispari $\Leftrightarrow n \equiv \pm 3 \pmod 8$.

3. $\frac{n^2-1}{8}$ è pari $\Leftrightarrow n \equiv \pm 1 \pmod 8 \Leftrightarrow$ è pari il numero di primi p_i congrui ± 3 modulo 8 (cioè tali che $\frac{p_i^2-1}{8}$ sono dispari) con e_i dispari.

4. Il numero di primi p_i congrui ± 3 modulo 8 (cioè tali che $\frac{p_i^2-1}{8}$ sono dispari) con e_i dispari è pari, se e solo se $\sum_{i=1}^{s} e_i \frac{p_i^2-1}{8}$ è pari.

Pertanto $\sum_{i=1}^{s} e_i \frac{p_i^2-1}{8}$ e $\frac{n^2-1}{8}$ hanno la stessa parità, e quindi si ha la tesi.

(c).

$$\left(\frac{m}{n}\right) = \prod_{j=1}^{s} \left(\frac{m}{p_j}\right)^{e_j} = \prod_{j=1}^{s} \prod_{i=1}^{r} \left(\frac{q_i}{p_j}\right)^{e_j f_i} = \prod_{j=1}^{s} \prod_{i=1}^{r} \left(\frac{p_j}{q_i}\right)^{e_j f_i} (-1)^{e_j f_i \frac{p_j-1}{2} \frac{q_i-1}{2}} =$$

$$= \left(\prod_{i=r}^{r} \left(\frac{n}{q_i}\right)^{f_i}\right) (-1)^{\sum_{j=1}^{s} e_j \frac{p_j-1}{2} \sum_{i=1}^{r} f_i \frac{q_i-1}{2}} = \left(\frac{n}{m}\right) (-1)^{\sum_{j=1}^{s} e_j \frac{p_j-1}{2} \sum_{i=1}^{r} f_i \frac{q_i-1}{2}}.$$

Per quanto detto nella (a), sappiamo che :

$\sum_{j=1}^{s} e_i \frac{p_j-1}{2} \equiv \frac{n-1}{2} \pmod 2$ cioè $\sum_{j=1}^{s} e_j \frac{p_j-1}{2} = \frac{n-1}{2} + 2k$ con $k \in \mathbb{Z}$

$\sum_{i=1}^{r} f_i \frac{q_i-1}{2} \equiv \frac{m-1}{2} \pmod 2$ cioè $\sum_{i=1}^{r} f_i \frac{q_i-1}{2} = \frac{m-1}{2} + 2h$ con $h \in \mathbb{Z}$

Quindi :

$$\sum_{j=1}^{s} e_j \frac{p_j-1}{2} \sum_{i=1}^{r} f_i \frac{q_i-1}{2} = \left(\frac{n-1}{2} + 2k\right)\left(\frac{m-1}{2} + 2h\right) =$$

$$= \frac{n-1}{2}\frac{m-1}{2} + 2\left(h\frac{n-1}{2} + k\frac{m-1}{2} + 2hk\right)$$

cioè

$$\sum_{j=1}^{s} e_j \frac{p_j-1}{2} \sum_{i=1}^{r} f_i \frac{q_i-1}{2} \equiv \frac{n-1}{2}\frac{m-1}{2} \pmod 2$$

□

Usando le proprietà del simbolo di Jacobi viste sopra, possiamo costruire il seguente algoritmo che consente di determinare il simbolo di Jacobi $\left(\frac{a}{n}\right)$ per qualsiasi coppia di interi positivi a e n, con n dispari.

Siano x e y interi positivi, con y dispari.

jacobi(x, y, t)

1. if ($x == 0$)
2. if ($y == 1$) then return t
3. else return **0**;
4. Calcola x', h' tali che $x = 2^{h'}x'$ e x' è dispari;

5. if ($h' \not\equiv 0 \pmod 2$ and $y \not\equiv \pm 1 \pmod 8$) then $t := -t$;

6. if ($x' \not\equiv 1 \pmod 4$ and $y \not\equiv 1 \pmod 4$) then $t := -t$;

7. return **jacobi**($y \bmod x'$, x', t)

L'algoritmo esposto corrisponde ad una funzione la cui legge di definizione è :

$$jacobi(x,y,t) = \begin{cases} 0 & \text{se } x = 0 \text{ e } y \neq 1 \\ t & \text{se } x = 0 \text{ e } y = 1 \\ t(-1)^{\frac{y^2-1}{8}h'}(-1)^{\frac{x'-1}{2}\frac{y-1}{2}} jacobi(y \bmod x', x', t) & \\ \text{con } x = 2^{h'}x', h' \geq 0 \text{ e } x'\text{dispari} & \text{altrimenti} \end{cases}$$

Infatti, a parte i primi due casi di definizione della funzione, banalmente identiche ai valori restituiti dall'algoritmo (linee 2 e 3), il terzo valore della funzione nasce da queste osservazioni :

1. Linea 5. Se h' è dispari e $y \not\equiv \pm 1 \pmod 8$ cioè $\frac{y^2-1}{8}$ è dispari e diverso da zero, allora il valore ottenuto dalla chiamata ricorsiva viene moltiplicato per -1. Questo equivale a moltiplicare il valore restituito dalla chiamata ricorsiva per $(-1)^{\frac{y^2-1}{8}h'}$.

2. Linea 6. Se $x' \not\equiv 1 \pmod 4$ e $y \not\equiv 1 \pmod 4$, cioè se $\frac{x'-1}{2}\frac{y-1}{2}$ è dispari, allora il valore ottenuto dalla chiamata ricorsiva viene moltiplicato per -1. Questo equivale a moltiplicare il valore restituito dalla chiamata ricorsiva per $(-1)^{\frac{x'-1}{2}\frac{y-1}{2}}$.

Teorema 53. *Sia a un intero positivo, allora per ogni n intero positivo e dispari si ha che $jacobi(a,n,1) = \left(\frac{a}{n}\right)$.*

Dimostrazione. Dimostriamo il teorema per induzione completa su a

Caso base : $a = 0$. Ci sono due sottocasi :

1. $n = 1$. In tal caso si ha $\left(\frac{0}{1}\right) = 1$ e $jacobi(0,1,1) = 1$.

2. $n \neq 1$. In tal caso si ha $\left(\frac{0}{n}\right) = 0$ e $jacobi(0,n,1) = 0$.

Passo induttivo : Siano a' un intero positivo dispari e h un intero positivo, tali che $a = 2^h a'$. Supponiamo, per ipotesi, che per ogni n intero positivo e dispari $jacobi(0, n, 1) = \left(\frac{0}{n}\right), ..., jacobi(2^h a' - 1, n, 1) = \left(\frac{2^h a' - 1}{n}\right)$ e dimostriamo che $jacobi(2^h a', n, 1) = \left(\frac{2^h a'}{n}\right) = \left(\frac{a}{n}\right)$.

$$\left(\frac{a}{n}\right) = \left(\frac{2^h a'}{n}\right) = \left(\frac{2^h}{n}\right)\left(\frac{a'}{n}\right) = \left(\frac{2}{n}\right)^h \left(\frac{a'}{n}\right) =$$

$$= (-1)^{\frac{n^2-1}{8}h}(-1)^{\frac{n-1}{2}\frac{a'-1}{2}}\left(\frac{n}{a'}\right) = (-1)^{\frac{n^2-1}{8}h}(-1)^{\frac{n-1}{2}\frac{a'-1}{2}}\left(\frac{n \bmod a'}{a'}\right)$$

Poichè $n \bmod a' \leq a' - 1 \leq 2^h a' - 1$, applicando l'ipotesi induttiva, otteniamo:

$$\left(\frac{a}{n}\right) = (-1)^{\frac{n^2-1}{8}h}(-1)^{\frac{n-1}{2}\frac{a'-1}{2}} jacobi(n \bmod a', a', 1) = jacobi(a, n, 1).$$

□

Parte II

Test algoritmici di primalità

Capitolo 3

Il Crivello di Eratostene

3.1 Il Crivello di Eratostene

Il crivello di Eratostene è paragonabile ad un setaccio che scartando tutti i numeri composti, permette di determinare i numeri primi. Data una sequenza di numeri 1, 2, ..., n il crivello procede nel seguente modo:

1. Si scrivono i numeri in ordine crescente
2. Si elimina l'uno (considerato non primo). La sua eliminazione è fondamentale per il funzionamento del crivello
3. Si incontra il 2, e si procede alla rimozione di tutti i multipli del 2, escluso il 2.
4. Si incontra il 3, e si procede alla rimozione di tutti i multipli del 3, escluso il 3.
5. In generale, se nella sequenza si incontra il numero i, si cancellano tutti i multipli di i, a partire dal numero $2i$.

Si dimostrerà che il setaccio, completa le sue operazioni quando elimina tutti i multipli di numeri primi minori o ugali a $\lfloor\sqrt{\text{n}}\rfloor$. Infine, dimostreremo la correttezza dell'algoritmo e la sua complessità.

3.2 L'Algoritmo

Sia $n \in N$ e A[2.. n] un array.

Eratostene(n)

1. for i = 2 to n do
2. $A[i] := 1$
3. for i = 2 to $\lfloor\sqrt{\text{n}}\rfloor$ do
4. if($A[i] == 1$) then

5. {

6. $j := 2 * i$

7. while($j \leq n$)

8. {

9. $A[j] := 0$

10. $j := j + i$

11. }

12. }

3.3 Dimostrazione dell'algoritmo

Lemma 7. *Sia $n \in \mathbb{N}\backslash\mathbb{P}$, allora $\exists p \in \mathbb{P}$ tale che $p \mid n$ e $p \leq \lfloor\sqrt{n}\rfloor$*

Dimostrazione. Se n è composto, consideriamo il più piccolo numero primo p che divide n. Sappiamo che $n = a \cdot p$ con $a \in \mathbb{N}$ e $a \geq p$. Infatti se fosse $a < p$ esisterebbe un primo $p' \in \mathbb{P}$ tale che $p' \mid a$. Cioè dimostreremo che $\exists p' \leq a < p$ primo e che divide n. Assurdo, perchè p è il più piccolo numero primo che divide n. Dunque, se $a \leq p$ abbiamo che :

$$n = a \cdot p \geq p \cdot p = p^2 \text{ da cui } p \leq \sqrt{n}$$

e poichè p è intero positivo si ha $p \leq \lfloor\sqrt{\mathrm{n}}\rfloor$. □

Teorema 54. *$A[i] = 1 \Leftrightarrow i$ è primo.*

Dimostrazione. Dimostriamo il teorema per induzione su i :

Caso base: verifichiamo la tesi per $i = 2$

Necessità : $A[2] = 1 \Rightarrow 2$ è primo (vero perchè 2 è primo)

Sufficienza : 2 è primo $\Rightarrow A[2] = 1$ (vero perchè l'algoritmo non modificherà mai $A[2]$ nel 2° ciclo for [linea 3]. Pertanto $A[2]$ manterrà il valore 1 che gli è stato attribuito nel

1° ciclo for [linea 1].

Passo induttivo : Supponiamo che il teorema sia vero per ogni numero minore o uguale a $i-1$ e dimostriamo il teorema per i.
Necessità : $A[i] = 1 \Rightarrow i$ primo.
Se $A[i] = 1$. Supponiamo per assurdo che i sia composto. Allora per il lemma $\exists p : p \mid i$ e $p \leq \lfloor\sqrt{\mathrm{n}}\rfloor \leq i-1$. Per ipotesi induttiva, poichè p è primo e $p \leq i-1$ vale il teorema, ovvero $A[p] = 1$. Ma se $A[p] = 1$, il ciclo while [linea 7] ci assicura che $A[2p] = 0, A[3p] = 0, ..., A[\frac{i}{p}p] = A[i] = 0$. Assurdo.
Sufficienza : se i è primo $\Rightarrow A[i] = 1$
Se i è primo. Supponiamo per assurdo che $A[i] = 0$. Affinchè $A[i]$ sia azzerato dall'algoritmo è necessario che $\exists a : 2 \leq a \leq \lfloor\sqrt{\mathrm{i}}\rfloor$ e $A[a] = 1$. Per ipotesi induttiva, se $A[a] = 1$, a sarà primo. Ma se a è primo e divide i allora i non è primo. Assurdo. □

3.4 Complessità dell'algoritmo

Per analizzare il tempo di esecuzione dell'algoritmo, assumiamo che le operazioni aritmetiche richiedano tempo costante. Ciò è ragionevole perchè tutte le quantità computate sono delimitate superiormente dal valore n. Una prima approssimazione, può essere ottenuta utilizzando il Teorema 39. Analizzando l'algoritmo notiamo che il ciclo while effettua al più $\frac{n}{i}$ azzeramenti nell'array $A[i]$ tanti quanti sono i multipli di i. Pertanto sommando tutti questi passi avremo :

$$\sum_{i\leq\sqrt{n}} \frac{n}{i} = n \sum_{i\leq\sqrt{n}} \frac{1}{i} = n \left[\int_1^{\sqrt{n}} \frac{1}{x}dx + \mathcal{O}(1)\right] \simeq n \left[\ln x\right]_1^{\sqrt{n}} \simeq n \ln(\sqrt{n}) \simeq n \ln n$$

Una approssimazione più dettagliata, può essere calcolata con il Teorema 41 :

$$\sum_{p\leq\sqrt{n}} \frac{n}{p} = n \sum_{p\leq\sqrt{n}} \frac{1}{p} = n \left[\ln\ln\sqrt{n} + \mathcal{O}(1)\right] = \mathcal{O}(n \ln\ln\sqrt{n})$$

$$= \mathcal{O}(n \ln(\frac{1}{2}\ln 2)) = \mathcal{O}(n \ln\frac{1}{2} + n \ln\ln n) = \mathcal{O}(n \ln\ln n)$$

Capitolo 4

L'algoritmo Miller-Rabin

4.1 Introduzione ai test probabilistici di primalità

Definizione 38 (Predicato n-ario o Problema di Decisione)**.** *Un predicato n-ario su* $\mathbb{N}$ *è una funzione* $P : \mathbb{N}^n \mapsto \{true, false\}$.

Ad ogni predicato possiamo associare una funzione caratteristica, definita nel seguente modo:

Definizione 39 (funzione caratteristica)**.** *Sia* P *un predicato n-ario, definiamo* funzione caratteristica di P *,e la denotiamo con* $C_P(\vec{x})$*, la funzione* $C_P : \mathbb{N}^n \mapsto \mathbb{Z}_2$ *la cui legge di definizione è:*

$$C_P(\vec{x}) = \begin{cases} 1 & se\ P(\vec{x}) = true \\ 0 & se\ P(\vec{x}) = false \end{cases}$$

Definire un predicato per il problema della primalità non è difficile. Possiamo, ad esempio, definire il predicato

$$Pr(n) : \text{“Il numero } n \text{ è composto”}.$$

Se il numero è composto, il predicato vale *true*, altrimenti *false*. Più difficile è costruire una procedura che valuti la funzione caratteristica associata al predicato. Una possibilità è quella di usare un algoritmo deterministico (come il crivello di Eratostene) che con una serie finita di passi, e senza far uso di numeri random, produce in output una risposta corretta. Un' altra possibilità è quella di usare un algoritmo probabilistico, che fa uso di numeri casuali e che per sua natura non consente di fornire una risposta certa. In questi casi ad ogni risposta è associato un grado di correttezza (o probabilità di errore). Definiamo dunque il concetto di test di primalità, come segue:

Definizione 40 (test di compositezza)**.** *Per un dato intero dispari n, definiamo* test di compositezza *su n, un predicato* $Pr(n, a)$*, con* $1 \leq a < n$ *, con le seguenti proprietà :*

1. n è composto se e solo se $\exists a : Pr(n, a) = true$
2. la funzione caratteristica C_{Pr} associata al predicato $Pr(n, a)$ è computabile in maniera efficiente.

3. Se n è composto, denotando con $L(n) = \{a \in \mathbb{Z}_n^+ : Pr(n,a) = \mathit{false}\}$, allora $|L(n)|/(n-1) \leq \epsilon$ con $0 < \epsilon \leq 1$.

Per un dato test, la quantità $\alpha_n = |L(n)|/(n-1)$ rappresenta la probabilità che un numero composto n sia dichiarato primo dal test e quindi ϵ è un upper bound a tale probabilità.
Inoltre, questa definizione ci suggerisce già un efficiente algoritmo probabilistico per testare la primalità. L' algoritmo consiste nel costruire una sequenza di m numeri interi random, $a_1, ..., a_m$ con $1 \leq a_i < n$. Se uno di questi numeri soddisfa il test allora n è composto. Altrimenti dichiariamo n primo ma con probabilità di errore minore o uguale a ϵ^m. Infatti la probabilità di errore dell'algoritmo, dopo m iterazioni, è $(\alpha_n)^m \leq \epsilon^m$. Un siffatto algoritmo appartiene alla classe degli algoritmi di tipo Montecarlo yes-biased.

Definizione 41 (algoritmi di tipo Montecarlo yes-biased)**.** *Un algoritmo di tipo Montecarlo yes-biased è un algoritmo probabilistico per un problema di decisione dove:*
- la risposta Yes è sempre corretta
- la risposta No può essere errata.

Ad ogni algoritmo di tipo montecarlo yes-biased è associata una probabilità di errore ϵ. Se l'algoritmo risponde No, allora con probabilità non superiore ad ϵ, la risposta sarà errata.

4.2 L'algoritmo di Miller-Rabin

4.2.1 I numeri di Carmichael

Abbiamo visto come il piccolo teorema di Fermat fornisce una condizione necessaria ma non sufficiente per la primalità di un numero. In particolare, se un numero n è primo, allora per ogni $a \in \mathbb{Z}_n^+$ si ha che $a^{n-1} \equiv 1 \, (mod\, n)$. Inoltre se n è primo sappiamo che $\mathbb{Z}_n^+ = \mathbb{Z}_n^*$. Ciò consente di riscrivere il teorema di Fermat nella forma $\forall n \in \mathbb{P}$ e $\forall a \in \mathbb{Z}_n^* \Rightarrow a^{n-1} \equiv 1 \pmod{n}$. Sfortunatamente esistono degli interi composti, detti *numeri di Carmichael*, definiti come segue:

Definizione 42. *Un intero composto n è detto di Carmichael se* $a^{n-1} \equiv 1 \pmod{n}\, \forall a \in \mathbb{Z}_n^*$.

Pertanto, il predicato

$$Pr_F(n, a) = \text{“}a^{n-1} \not\equiv 1\ (mod\, n)\text{”}$$

non può essere assunto come criterio per definire un test di compositezza a tutti gli effetti. Per definizione di test di compositezza, si dovrebbebbero verificare entrambe le condizioni:

$$\exists a \in \mathbb{Z}_n^+,\ Pr_F(n, a) = true \Rightarrow n \in \mathbb{N}\backslash\mathbb{P}$$

$$n \in \mathbb{N}\backslash\mathbb{P} \Rightarrow \exists a \in \mathbb{Z}_n^+,\ Pr_F(n, a) = true$$

Ma, delle due, solo la prima è certamente vera (è un modo diverso di esprimere il teorema di Fermat). La seconda è quasi sempre valida. Perciò, assumiamo $Pr_F(n, a)$ come test di pseudoprimalità. Ovvero fissato un n e scelto un a, se $Pr_F(n, a) = true$, dichiariamo con certezza che n è composto. Altrimenti, se $Pr_F(n, a) = false$, n è un primo o uno pseudoprimo di base a.

Definizione 43. *Un intero composto n è uno* pseudoprimo di base a *con* $a \in \mathbb{Z}_n^+$ *se* $a^{n-1} \equiv 1 \pmod{n}$

Il teorema che segue fornisce un limite superiore al valore di α_n, sotto ipotesi strette, e nel caso in cui si assume il predicato Pr_F come criterio per definire il test di primalità.

Definizione 44. $L_F(n) = \{a \in \mathbb{Z}_n^+ : a^{n-1} \equiv 1\ (mod\, n)\}$

Teorema 55. *Se n è primo, allora $L_F(n) = \mathbb{Z}_n^*$. Se n è composto e $L_F(n) \subset \mathbb{Z}_n^*$, allora* $\mid L_F(n) \mid \leq (n-1)/2$.

Dimostrazione. Se n è primo, per il teorema di Fermat, $L_F(n) = \mathbb{Z}_n^+$. Ma poichè quando n è primo si ha anche $\mathbb{Z}_n^+ = \mathbb{Z}_n^*$ segue che $L_F(n) = \mathbb{Z}_n^*$. Supponiamo che n sia composto e $L_F(n) \subset \mathbb{Z}_n^*$. Definito l'omomorfismo $f : \mathbb{Z}_n^* \to \mathbb{Z}_n^*$ tale che $f(a) = a^{n-1} \mod n$, si ha che $Ker(f) = L_F(n)$ e quindi $L_F(n)$ è un sottogruppo di $\mathbb{Z}_n^*$. A questo punto, poichè la

cardinalità di un sottogruppo divide la cardinalità del gruppo, possiamo scrivere che : $\mid \mathbb{Z}_n^* \mid = m \mid L_F(n) \mid$ per qualche intero $m > 1$, da cui, concludiamo che

$$\mid L_F(n) \mid = \frac{1}{m} \mid \mathbb{Z}_n^* \mid \leq \frac{1}{2} \mid \mathbb{Z}_n^* \mid \leq \frac{n-1}{2}.$$

□

Mostrerò, in seguito, come migliorare il predicato $Pr_F(n, a)$ per far sì che la verifica della primalità non fallisca con i numeri di Carmichael. Prima di fare ciò, è utile citare qualche proprietà di questi numeri:

Teorema 56. *Un numero di Carmichael è della forma $n = p_1 \cdots p_r$, dove i $p_i \in \mathbb{P}$ sono a due a due distinti, $r \geq 3$, e $(p_i - 1) \mid (n-1) \forall i = 1, ..., r$.*

Dimostrazione. Osserviamo, per iniziare, che per ogni $n \in \mathbb{Z}^+$ vale $L_F(n) \subseteq \mathbb{Z}_n^*$. Infatti se $a^{n-1} \equiv 1 \pmod{n}$, a ha come inverso moltiplicativo modulo n il numero a^{n-2}. Quindi $\gcd(a, n) = 1$, da cui $a \in \mathbb{Z}_n^*$. Tenendo conto che la cardinalità dell'insieme $L_F(n)$ si può calcolare dal Teorema 26 e che n è di Carmichael se e solo se $\forall a \in \mathbb{Z}_n^*, a \in L_F(n)$, o equivalentemente $L_F(n) = \mathbb{Z}_n^*$, passando alle cardinalità, possiamo dire che n della forma $p_1^{a_1} \cdot .... \cdot p_r^{a_r}$ è di Carmichael se e solo se

$$\prod_{1 \leq i \leq r} \gcd(n-1, \phi(p_i^{a_i})) = \mid L_F(n) \mid = \mid \mathbb{Z}_n^* \mid = \phi(n) = \prod_{1 \leq i \leq r} \phi(p_i^{a_i}) \ \forall 1 \leq i \leq r.$$

Ma ciò si verifica se e solo se :

$$\phi(p_i^{a_i}) \mid n-1 \ \forall 1 \leq i \leq r.$$

Pertanto, se $n = p_1^{a_1} \cdot .... \cdot p_r^{a_r}$ è un numero di Carmichael, allora $\phi(p_i^{a_i}) \mid n-1$, per $1 \leq i \leq r$. Ma se $a_i > 1$, ricordando che $\phi(p_i^{a_i}) = p_i^{a_i - 1}(p_i - 1)$, si avrebbe che $p_i \mid n-1$. Assurdo, in quanto $p_i \mid n$. Ciò implica che $a_1 = a_2 = ... = a_r = 1$, cioè $n = p_1 \cdot p_2 \cdot .... \cdot p_r$. Ci rimane di dimostrare che $r \geq 3$. Supponiamo per assurdo che $r = 2$, così che $n = p_1 p_2$. Quindi :

$$n - 1 = p_1 p_2 - 1 = (p_1 - 1)p_2 + (p_2 - 1).$$

Poichè $(p_1 - 1) \mid (n-1)$, si ha che $(p_1 - 1) \mid (p_2 - 1)$. Con un ragionamento analogo, $(p_2 - 1) \mid (p_1 - 1)$. Quindi, $p_1 = p_2$. Assurdo. □

Infine, detto $C(x)$ il numero dei numeri di Carmichael minori o uguali ad x, Alford, Granville e Pomerance, hanno dimostrato nel 1994 che $C(x) > x^{2/7}$ per x sufficientemente grande. Questo teorema, di fatto, implica che i numeri di Carmichael sono infiniti, e quindi che è impossibile costruire un algoritmo che testi la primalità di un numero con:

1. un controllo della condizione del piccolo teorema di Fermat
2. una verifica dell'appartenenza o meno del numero in questione alla lista di tutti i numeri di Carmichael.

4.2.2 L'Idea dell'Algoritmo di Miller-Rabin

L'idea che sta dietro all'algoritmo di Miller-Rabin è quella di migliorare il controllo basato solo sul predicato $Pr_F(n, a)$, e può essere riassunta nel seguente punto : Mentre si calcola l'elevamento a potenza modulare a^{n-1}, si controlla se esistono radici quadrate non banali di 1 modulo n. Se ciò accade, si interrompe l'algoritmo e si segnala che n è composto.

Ecco dunque come conviene modificare il predicato $Pr_F(n, a)$, dando luogo ad un nuovo predicato $Pr'_{MR}(n, a)$:

Definizione 45. *Sia $n \in \mathbb{N}$ dispari e siano $m_0, \nu_0 \in \mathbb{N}$ tali che $n - 1 = 2^{\nu_0} m_0$ con m_0 dispari. Poniamo :*

$Pr'_{MR}(n, a) =$ "$a^{n-1} \not\equiv 1 \pmod{n} \vee$
$\exists k : 0 \leq k < \nu_0,$ *tale che* $(a^{m_0 2^{k+1}} \equiv 1 \pmod{n} \wedge a^{m_0 2^k} \not\equiv \pm 1 \pmod{n})$"

Nella pratica però non si adotta il predicato $Pr'_{MR}(n, a)$ per costruire l'algoritmo, ma un predicato ad esso equivalente, che chiamiamo $Pr_{MR}(n, a)$, così definito :

Definizione 46. *Sia $n \in \mathbb{N}$ dispari e siano $m_0, \nu_0 \in \mathbb{N}$ tali che $n - 1 = 2^{\nu_0} m_0$ con m_0 dispari. Poniamo :*

$Pr_{MR}(n,a) = \neg$ "$C_{-1}(n,a) \vee \exists j_0$ *con* $0 \leq j_0 < \nu_0 : C_{j_0}(n,a)$)"

dove :

$C_{-1}(n,a) =$ "$a^{m_0} \equiv 1 \pmod{n}$"

$C_j(n,a) =$ "$a^{m_0 2^j} \equiv -1 \pmod{n}$", $j \in \mathbb{N}$

Dimostriamo, adesso, che i due predicati $Pr_{MR}(n,a)$ e $Pr'_{MR}(n,a)$ sono equivalenti.

Teorema 57. $Pr'_{MR}(n,a) \Leftrightarrow Pr_{MR}(n,a)$

Dimostrazione. Per dimostrare il teorema conviene esplicitare le negazioni dei due predicati $Pr_{MR}(n,a)$ e $Pr'_{MR}(n,a)$:
$\neg Pr_{MR}(n,a) = C_{-1}(n,a) \vee \exists j_0$ con $1 \leq j_0 < \nu_0 : C_{j_0}(n,a)$
$\neg Pr'_{MR}(n,a) =$ "$a^{n-1} \equiv 1 \pmod{n} \wedge \forall k : 0 \leq k < \nu_0, a^{m_0 2^{k+1}} \equiv 1 \pmod{n} \Rightarrow a^{m_0 2^k} \equiv \pm 1 \pmod{n}$".

Necessità: ($\neg Pr_{MR}(n,a) \Rightarrow \neg Pr'_{MR}(n,a)$) :
Si possono verificare i seguenti due casi :

1. Vale $C_{-1}(n,a)$, cioè $a^{m_0} \equiv 1 \pmod{n}$. In questo caso si ha $1 \equiv a^{m_0} \equiv a^{2m_0} \equiv ... \equiv a^{2^{\nu_0} m_0} \equiv a^{n-1} \pmod{n}$ pertanto $\neg Pr'_{MR}(n,a)$.

2. Vale $C_{j_0}(n,a)$ cioè $a^{m_0 2^{j_0}} \equiv -1 \pmod{n}$, per qualche $0 \leq j_0 < \nu_0$. In tal caso si ha $1 \equiv a^{m_0 2^{j_0+1}} \equiv ... \equiv a^{2^{\nu_0} m_0} \equiv a^{n-1} \pmod{n}$ e $a^{m_0 2^j} \not\equiv \pm 1 \pmod{n}$, per ogni $0 \leq j < j_0$. Pertanto il predicato $\neg Pr'_{MR}(n,a)$ risulta vero.

Sufficienza: ($\neg Pr'_{MR}(n,a) \Rightarrow \neg Pr_{MR}(n,a)$) :
Per ipotesi sappiamo che $a^{n-1} \equiv a^{m_0 2^{\nu_0}} \equiv 1 \pmod{n}$ e che vale $a^{m_0 2^{k+1}} \equiv 1 \pmod{n} \Rightarrow a^{m_0 2^k} \equiv \pm 1 \pmod{n}, \forall k : 0 \leq k < \nu_0$. Chiaramente se $C_{-1}(n,a)$ vale, il predicato $\neg Pr_{MR}(n,a)$ risulta verificato. Supponiamo quindi che $\neg C_{-1}(n,a)$ sia vero, cioè che $a^{m_0} \equiv 1 \pmod{n}$. Se vale $\neg C_{j_0}(n,a)$, per ogni $0 \leq j_0 < \nu_0$ cioè se $a^{m_0 2^{j_0}} \not\equiv -1 \pmod{n}$ per ogni $0 \leq j_0 < \nu_0$, allora per quanto osservato sopra si ha: $1 \equiv a^{n-1} \equiv a^{m_0 2^{\nu_0}} \equiv a^{m_0 2^{\nu_0 - 1}} \equiv ... \equiv a^{m_0 \cdot 2} \equiv a^{m_0} \pmod{n}$, il che è assurdo in quanto abbiamo supposto che

$a^{m_0} \not\equiv 1 \pmod{n}$. Pertanto $C_{j_0}(n, a)$ deve essere vero per qualche $0 < j_0 \leq \nu_0$, da cui segue $\neg Pr_{MR}(n, a)$. □

4.2.3 L'Algoritmo

Sia $n \in \mathbb{N}$ dispari e siano $m_0, \nu_0 \in \mathbb{N}$ tali che $n - 1 = 2^{\nu_0} m_0$ con m_0 dispari

Miller-Rabin$(n,\ s)$

1. for $i = 1$ to s
2. if (test-MR(n)) then return **composto** ;
3. return **primo**

---

test-MR(n)

1. $a :=$ Random$(1,\ n-1)$
2. $x := a^{m_0} \mod n$
3. if$(x \equiv 1 \pmod{n})$ then return **false**;
4. for j $= 0$ to $\nu_0 - 1$ do
5. if ($x \equiv -1 \pmod{n}$) then return **false**
6. $x := x^2 \mod n$;
7. return **true**

4.2.4 Dimostrazione della correttezza dell'algoritmo di Miller-Rabin

Dimostrare la correttezza dell'algoritmo di Miller-Rabin, significa dimostrare che il predicato $Pr_{MR}(n, a)$ soddisfa le condizioni per potersi definire test probabilistico di compositezza, in particolare che $n \in \mathbb{N}\backslash\mathbb{P} \iff \exists a \in \mathbb{Z}_n^+ : Pr_{MR}(n, a) = true$.

Teorema 58. $n \in \mathbb{N}\backslash\mathbb{P} \Longleftrightarrow \exists a \in \mathbb{Z}_n^+ : Pr_{MR}(n,a) = true$

Dimostrazione. Necessità : $n \in \mathbb{N}\backslash\mathbb{P} \Rightarrow \exists a \in \mathbb{Z}_n^+ : Pr_{MR}(n,a) = true$
Dimostrare la necessità, equivale a dimostrare che $n \in \mathbb{N}\backslash\mathbb{P} \Rightarrow \exists a \in \mathbb{Z}_n^+ : Pr'_{MR}(n,a) = true$. Introducendo l'insieme :

$L'_{MR}(n) = \{a \in \mathbb{Z}_n^+ : \neg Pr'_{MR}(n,a)\} =$
$= \{a \in \mathbb{Z}_n^+ : a^{m_0 2^{\nu_0}} \equiv 1 \pmod{n} \wedge \forall k : 0 \leq k < \nu_0,\ a^{m_0 2^{k+1}} \equiv 1 \pmod{n} \Rightarrow a^{m2^k} \equiv \pm 1 \pmod{n}\}$,

per dimostrare la necessità basta fare vedere che se $n \in \mathbb{N}\backslash\mathbb{P}$ allora $L'_{MR}(n) \neq \mathbb{Z}_n^+$.
Dimostreremo ciò facendo vedere che la probabilità di trovare un testimone del fatto che n è composto è maggiore di zero. Cioè supposto n composto allora $P[Pr'_{MR}(n,a) = true] > 0$. Dalla probabilità dell'evento contrario, abbiamo che :

$$P[Pr'_{MR}(n,a) = true] = 1 - P[Pr'_{MR}(n,a) = false] = 1 - \alpha_n = 1 - \frac{\mid L'_{MR}(n) \mid}{n-1}.$$

Se n è composto, per il Teorema 59 che sarà dimostrato nel paragrafo successivo, si ha che $\frac{|L'_{MR}(n)|}{n-1} \leq 1/4$, da cui :

$$P[Pr'_{MR}(n,a) = true] \geq 1 - 1/4 = 3/4 > 0.$$

Sufficienza : $\exists a \in \mathbb{Z}_n^+ : Pr_{MR}(n,a) = true \Rightarrow n \in \mathbb{N}\backslash\mathbb{P}$
Siano $n \in \mathbb{N}, a \in \mathbb{Z}_n^+$ tali che $Pr'_{MR}(n,a) = true$ e supponiamo per assurdo che $n \in \mathbb{P}$. Poichè il predicato $Pr'_{MR}(n,a)$ è vero, possiamo distinguere i seguenti due casi :

1. $a^{n-1} \not\equiv 1 \pmod{n}$. In questo caso poichè stiamo supponendo n primo, per il piccolo teorema di Fermat $a^{n-1} \equiv 1 \pmod{n}$. Assurdo.

2. $\exists k : 0 \leq k < \nu_0$, tale che $a^{m_0 2^{k+1}} \equiv 1 \pmod{n} \wedge a^{m_0 2^k} \not\equiv \pm 1 \pmod{n}$. Cioè $a^{m_0 2^{k+1}}$ ha una radice quadrata non banale di 1 modulo n. E quindi per il teorema 27, n dovrebbe essere composto. Assurdo.

□

4.2.5 Probabilità di errore del Test di Miller-Rabin

Teorema 59. *Se n è primo, allora $L'_{MR}(n) = \mathbb{Z}_n^*$. Se n è un numero dispari e composto, allora $\mid L'_{MR}(n) \mid \leq (n-1)/4$.*

Dimostrazione. Caso 1 : n è primo. Sappiamo che $L'_{MR}(n) \subseteq \mathbb{Z}_n^+ = \mathbb{Z}_n^*$. Dunque, per dimostrare la tesi, basta dimostrare che $\mathbb{Z}_n^* \subseteq L'_{MR}(n)$. Sia dunque $a \in \mathbb{Z}_n^*$. Per il teorema di fermat, $a^{m_0 2^{\nu_0}} = a^{n-1} \equiv 1 \pmod{n}$. Inoltre, se $a^{m_0 2^{k+1}} \equiv 1 \pmod{n}$ per qualche $0 \leq k < \nu_0$, poichè n è primo si ha che $a^{m_0 2^k} \equiv \pm 1 \pmod{n}$ [per il corollario 4]. Pertanto $a \in L'_{MR}(n)$ e quindi $\mathbb{Z}_n^* \subseteq L'_{MR}(n)$.

Caso 2 : $n = p^a$, dove p è primo e $a > 1$. Definito l'omomorfismo $\rho : \mathbb{Z}_n^* \to \mathbb{Z}_n^*$ come $\rho(a) = a^{n-1} mod\ n$, si ha che $Ker(\rho) \subseteq \mathbb{Z}_n^*$. Dal Teorema 26, otteniamo che $\mid Ker(\rho) \mid = \gcd(\phi(n), n-1)$. E quindi:

$$\mid L'_{MR}(n) \mid \leq \mid Ker(\rho) \mid = \gcd(\phi(n), n-1) =$$
$$= \gcd(p^{a-1}(p-1), p^a - 1) = p - 1 = \frac{p^a - 1}{p^{a-1} + ... + 1} \leq \frac{n-1}{4}.$$

Caso 3 : $n = p_1^{a_1} \cdot p_2^{a_2} \cdots p_r^{a_r}$, con $r > 1$. Sia

$$\theta : \mathbb{Z}_{p_1^{a_1}}^* \times \cdots \times \mathbb{Z}_{p_r^{a_r}}^* \to \mathbb{Z}_n^*$$

l'isomorfismo analogo a quello utilizzato nella dimostrazione del teorema cinese del resto. E inoltre siano $\phi(p_i^{a_i}) = m_i 2^{\nu_i}$, con m_i dispari, per ogni $i = 1, ..., r$ e sia $l = min\{\nu_0, \nu_1, ..., \nu_r\}$. Notiamo che $l \geq 1$ e che ciascun $\mathbb{Z}_{p_i^{a_i}}^*$ è ciclico di ordine $m_i 2^{\nu_i}$. Iniziamo col dimostrare che per ogni $a \in L'_{MR}(n)$, si ha che $a^{m_0 2^l} \equiv 1 \pmod{n}$. A questo fine, notiamo che se $l = \nu_0$, allora per definizione di $n-1$, si ha che $a^{m_0 2^l} \equiv 1 \pmod{n}$. Adesso dimostriamo che la tesi vale anche per $l < \nu_0$, nel qual caso si avrà che $l = \nu_i$ per qualche $i = 1, ..., r$. Supponiamo per assurdo, che $a^{m_0 2^l} \not\equiv 1 \pmod{n}$ e sia j il più piccolo indice nel range $[l..\nu_0 - 1]$ tale che $a^{m_0 2^{j+1}} \not\equiv 1 \pmod{n}$. Dalla definizione

di $L'_{MR}(n)$ si ha che $a^{m_0 2^j} \equiv -1 \pmod{n}$. Scrivendo a come $a = \theta(\alpha_1, ..., \alpha_r)$, abbiamo che $\alpha_i^{m_0 2^j} \equiv -1 \pmod{p_i^{a_i}}$. Pertanto, per il Teorema 16, l'ordine dell'elemento $\alpha_i^{m_0}$ è uguale a 2^{j+1}. Ma l'ordine di un elemento di un gruppo (in questo caso $\alpha_i^{m_0}$) dovrebbe dividere l'ordine del gruppo (in questo caso, $\mathbb{Z}^*_{p_i^{a_i}}$). Quindi, $2^{j+1} \mid 2^{\nu_i}$, da cui $j+1 \leq \nu_i$, il che è assurdo in quanto $j \geq l = \nu_i$.
A questo punto, per la definizione di $L'_{MR}(n)$, se $a \in L'_{MR}(n)$ allora $a^{m_0 2^{l-1}} \equiv \pm 1 \pmod{n}$. Supponiamo che i numeri casuali a usati dall'algoritmo di Miller-Rabin siano estratti in maniera casuale da $\mathbb{Z}^*_n$ con distribuzione uniforme di probabilità e dimostriamo che la probabilità $P[a^{m_0 2^{l-1}} \equiv \pm 1 \pmod{n}] \leq 1/4$, da cui segue il nostro teorema. Infatti, $P[a^{m_0 2^{l-1}} \equiv \pm 1 \pmod{n}] \leq \frac{1}{4}$ implica che $\mid \{a \in \mathbb{Z}^*_n : a^{m_0 2^{l-1}} \equiv \pm 1 \pmod{n}\} \mid \leq \frac{\phi(n)}{4} \leq \frac{n-1}{4}$ e poiché $L'_{MR}(n) \subseteq \{a \in \mathbb{Z}^+_n : a^{m_0 2^{l-1}} \equiv \pm 1 \pmod{n}\} \subseteq \{a \in \mathbb{Z}^*_n : a^{m_0 2^{l-1}} \equiv \pm 1 \pmod{n}\}$, allora $P[a^{m_0 2^{l-1}} \equiv \pm 1 \pmod{n}] \leq \frac{1}{4}$ implica $\mid L'_{MR}(n) \mid \leq \frac{n-1}{4}$. Dimostriamo pertanto che $P[a^{m_0 2^{l-1}} \equiv \pm 1 \pmod{n}] \leq \frac{1}{4}$.

Scriviamo a nella forma $a = \theta(\alpha_1, ..., \alpha_r)$ con gli α_i scelti con distribuzione di probabilità uniforme su $\mathbb{Z}^*_{p_i^{a_i}}$. Per ogni $i = 1, ..., r$ e $j = 0, ..., \nu_0$ definiamo le funzioni $\rho_{i,j} : \mathbb{Z}^*_{p_i^{a_i}} \to \mathbb{Z}^*_{p_i^{a_i}}$ come $\rho_{i,j}(a) = a^{m_0 2^j} \mod p_i^{a_i}$ e indichiamo con $G_i(j) = Im(\rho_{i,j})$. Sappiamo che $\mid \mathbb{Z}^*_{p_i^{a_i}} \mid = \phi(p_i^{a_i}) = m_i 2^{\nu_i}$. Scelto un generatore $a \in \mathbb{Z}^*_{p_i^{a_i}}$, sia $b = \rho_{i,j}(a) \in G_i(j)$. Quindi per il Teorema 17 :

$$\mid G_i(j) \mid = ord(b) = ord(a^{m_0 2^j}) = \frac{m_i 2^{\nu_i}}{\gcd(m_i 2^{\nu_i}, m_0 2^j)}.$$

Poichè $l \leq \nu_0$ e $l \leq \nu_i$, notiamo che :

$$\mid G_i(\nu_0) \mid \text{ divide } \mid G_i(l) \mid \text{ e } 2 \mid G_i(l) \mid = \mid G_i(l-1) \mid .$$

In particolare $\mid G_i(l-1) \mid \geq 2 \mid G_i(\nu_0) \mid$. Ma l'evento $a^{m_0 2^{l-1}} \equiv \pm 1 \pmod{n}$ si verifica se e solo se si verifica uno di questi eventi (dall'isomorfismo definito nel teorema cinese del resto) :
(E_1) $\alpha_i^{m_0 2^{l-1}} \equiv 1 \pmod{p_i^{a_i}}$ $\forall i = 1, ..., r$ oppure
(E_2) $\alpha_i^{m_0 2^{l-1}} \equiv -1 \pmod{p_i^{a_i}}$ $\forall i = 1, ..., r$.

Questi eventi sono disgiunti, i valori $\alpha_i^{m_0 2^{l-1}}$ sono scelti con distribuzione di probabilità uniforme su $G_i(l-1)$, e quindi possiamo calcolare la probabilità:

$$P[a^{m_0 2^{l-1}} \equiv \pm 1 \pmod{n}] = P[E_1] + P[E_2] = 2\prod_{i=1}^{r} \frac{1}{\mid G_i(l-1) \mid}.$$

Ricordando che $\mid G_i(l-1) \mid \geq 2 \mid G_i(\nu_0) \mid$, otteniamo :

$$\frac{1}{\mid G_i(l-1) \mid} \leq \frac{1}{2 \mid G_i(\nu_0) \mid}$$

$$\prod_{i=1}^{r} \frac{1}{\mid G_i(l-1) \mid} \leq \prod_{i=1}^{r} \frac{1}{2 \mid G_i(\nu_0) \mid}$$

$$2\prod_{i=1}^{r} \frac{1}{\mid G_i(l-1) \mid} \leq 2\prod_{i=1}^{r} \frac{1}{2 \mid G_i(\nu_0) \mid}$$

$$2\prod_{i=1}^{r} \frac{1}{\mid G_i(l-1) \mid} \leq 2\frac{1}{2^r}\prod_{i=1}^{r} \frac{1}{\mid G_i(\nu_0) \mid}$$

Concludiamo che :

$$P[a^{m_0 2^{l-1}} \equiv \pm 1 \pmod{n}] \leq 2^{-r+1}\prod_{i=1}^{r} \frac{1}{\mid G_i(\nu_0) \mid}.$$

Se $r \geq 3$ allora, direttamente notiamo che $P[a^{m_0 2^{l-1}} \equiv \pm 1 \pmod{n} \leq 1/4$.
Se $r = 2$ allora n non è un numero di Carmichael. Ciò implica che $\exists \overline{a} \in \mathbb{Z}_n^* : a^{n-1} \not\equiv 1 \pmod{n}$, per qualche $\overline{a} \in \mathbb{Z}_n^*$. Dimostriamo adesso che $\exists i \in \{1, 2\}$ tale che $\mid G_i(\nu_0) \mid \geq 2$. Ovviamente $1 \in G_1(\nu_0)$ e $1 \in G_2(\nu_0)$. Supponiamo per assurdo che $G_1(\nu_0) = G_2(\nu_0) = \{1\}$. Allora :

$$\forall \alpha_1 \in \mathbb{Z}_{p_1^{a_1}}^* \text{ si ha } \alpha_1^{n-1} \equiv 1 \pmod{p_1^{a_1}}$$

$$\forall \alpha_2 \in \mathbb{Z}_{p_2^{a_2}}^* \text{ si ha } \alpha_2^{n-1} \equiv 1 \pmod{p_2^{a_2}}.$$

Adesso, scelti $\overline{\alpha}_1$ e $\overline{\alpha}_2$ in modo tale che $\overline{a} = \theta(\overline{\alpha}_1, \overline{\alpha}_2)$, per il teorema cinese del resto si ha che $\overline{a}^{n-1} \equiv 1 \pmod{n}$, il che è assurdo. Quindi, $\exists i \in \{1, 2\}$ tale che $\mid G_i(\nu_0) \mid \geq 2$ e pertanto

$$P[a^{m_0 2^{l-1}} \equiv \pm 1 \pmod{n}] \leq \frac{1}{2} \cdot \frac{1}{\mid G_1(\nu_0) \mid} \cdot \frac{1}{\mid G_2(\nu_0) \mid} \leq 1/4.$$

Ciò conclude la dimostrazione del teorema. □

4.2.6 Complessità dell'algoritmo

Analizzando l'algoritmo, notiamo che il test di Miller-Rabin viene ripetuto s volte. Ogni test effettua :

1. un elevamento a potenza modulare nella linea 2 : Costo computazionale con l'algoritmo square and multiply $\mathcal{O}((\log n)^3)$

2. Al più $\nu_0 < \log n$ operazioni aritmetiche : Ciascuna di queste, richiede un costo computazionale pari a $\mathcal{O}((\log n)^2)$

Per un totale di $\mathcal{O}(s(\nu_0(\log n)^2 + \mathcal{O}((\log n)^3))$. Concludiamo che l'algoritmo di Miller-Rabin impiega $\mathcal{O}(s(\log n)^3)$.

Capitolo 5

L'algoritmo Solovay-Strassen

5.1 L'idea dell'algoritmo di Solovay-Strassen

La definizione del simbolo di Jacobi e il criterio di Eulero, suggeriscono la costruzione di un algoritmo probabilistico che permette di testare la compositezza di un numero. Ricordiamo, infatti, che se n è primo, il simbolo di jacobi $\left(\frac{a}{n}\right)$ è equivalente al simbolo di Legendre e grazie al criterio di Eulero, sappiamo calcolarlo in maniera efficiente. Inoltre, ancora per il criterio di Eulero, se n è primo si ha che $\left(\frac{a}{n}\right) \equiv a^{\frac{n-1}{2}} \pmod{n} \forall a \in \mathbb{Z}$. Quindi, esiste un $a \in \mathbb{Z} : \left(\frac{a}{n}\right) \not\equiv a^{\frac{n-1}{2}} \pmod{n}$ allora, con certezza, n è composto. D'altra parte, se per qualche $a \in \mathbb{Z}, \left(\frac{a}{n}\right) \equiv a^{\frac{n-1}{2}} \pmod{n}$ non è detto che n sia primo. Introduciamo, dunque, questa definizione :

Definizione 47. *Un intero dispari n si dice* pseudoprimo di Eulero *rispetto ad una base a tale che* $\gcd(a, n) = 1$ *se*

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

Per quanto osservato prima, se n è primo allora n è pseudoprimo di Eulero rispetto ad ogni base a. Inoltre se n è pseudoprimo di Eulero rispetto alla base a, allora n è pseudoprimo di Fermat rispetto alla stessa base, ma non vale il viceversa. Infatti :

$$a^{\frac{n-1}{2}} \equiv \left(\frac{a}{n}\right) \pmod{n} \Rightarrow a^{\frac{n-1}{2}} \equiv (\pm 1)^2 \pmod{n} \Rightarrow a^{n-1} \equiv 1 \pmod{n}.$$

L'idea dell'algoritmo sarà quella di testare, più volte e per basi a scelte in maniera random, la condizione di Eulero espressa dal predicato:

$$Pr_{SS}(n, a) = \text{“}\left(\frac{a}{n}\right) \not\equiv a^{\frac{n-1}{2}} \pmod{n} \text{ e } \gcd(a, n) = 1\text{”}.$$

Se vale il predicato allora n sarà dichiarato *composto* con certezza, altrimenti n sarà dichiarato *pseudoprimo* di Eulero rispetto alla base a.

5.2 L'algoritmo

Sia n un intero dispari e s un numero naturale.

Solovay-Strassen(n, s)

1. for $i = 1$ to s do
2. if (test-SS(n)) then return **composto** ;
3. return **primo**

---

test-SS(n)

1. $a :=$ Random(2, $n-2$);
2. if ($\gcd(a, n) > 1$) then return **true**;
3. if ($\left(\frac{a}{n}\right) \not\equiv a^{\frac{n-1}{2}} \pmod{n}$) then return **true**;
4. else return **false**;

5.3 Dimostrazione della correttezza dell'algoritmo di Solovay-Strassen

Dimostrare la correttezza dell'algoritmo di Solovay Strassen, significa dimostrare che il predicato $Pr_{SS}(n, a)$ soddisfa le condizioni per potersi definire test probabilistico di compositezza, in particolare che $n \in \mathbb{N}\backslash\mathbb{P} \Leftrightarrow \exists a \in \mathbb{Z}_n^+ : Pr_{SS}(n, a) = true$.

Teorema 60. *Sia n un numero naturale dispari. Si ha:*

$$n \in \mathbb{N}\backslash\mathbb{P} \Leftrightarrow \exists a \in \mathbb{Z}_n^+ : Pr_{SS}(n, a) = true.$$

Dimostrazione. Necessità : $n \in \mathbb{N}\backslash\mathbb{P} \Rightarrow \exists a \in \mathbb{Z}_n^+ : \left(\frac{a}{n}\right) \not\equiv a^{\frac{n-1}{2}} \pmod{n}$ e $\gcd(a, n) = 1$
Sia $n = p_1 p_2 ... p_k$ la fattorizzazione di n come prodotto di primi. Distinguiamo i due seguenti casi.

1. Esiste un primo p tale che $p^2 \mid n$, cioè $p = p_l = p_m$ con $1 \leq l < m \leq k$. Scelto $a = 1 + \frac{n}{p}$, dimostriamo che $Pr_{SS}(n, a) = true$. Innanzitutto, dato che $a \equiv 1 \pmod{p_i} \forall i = 1...k$, deduciamo che :

$$\left(\frac{a}{n}\right) = \prod_{i=1}^{k} \left(\frac{a}{p_i}\right) = \prod_{i=1}^{k} \left(\frac{1}{p_i}\right) = 1.$$

Inoltre per ogni $g \in \mathbb{N}$:

$$a^g = \left(1 + \frac{n}{p}\right)^g = \sum_{j=0}^{g} \binom{g}{j} \left(\frac{n}{p}\right)^j = 1 + \binom{g}{1}\left(\frac{n}{p}\right) + n\sum_{j=2}^{g} \binom{g}{j}\left(\frac{n^{j-1}}{p^j}\right).$$

Poichè $p^2 \mid n$ allora $\sum_{j=2}^{g} \binom{g}{j}\left(\frac{n^{j-1}}{p^j}\right) \in \mathbb{N}$, dunque $a^g \equiv 1 + g\frac{n}{p} \pmod{n}$. Ma a^g è congruo 1 modulo n solo quando $p \mid g$ e poichè p non divide $\frac{n-1}{2}$ si ha $a^{\frac{n-1}{2}} \not\equiv 1 \pmod{n}$. Infine è immediato notare che $\gcd(1 + \frac{n}{p}, n) = 1$.

2. Sia $p_1 p_2 ... p_k$ il prodotto di primi distinti. In tal caso è sufficiente trovare un a tale che :

$$\begin{cases} \left(\frac{a}{p_i}\right) = -1 \\ a \equiv 1 \pmod{n/p_i} \end{cases}$$

per qualche p_i con $i = 1...k$. Infatti, poichè $a \equiv 1 \pmod{p_j} \forall j \neq i$, si ha

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_i}\right) \prod_{\substack{j=0 \\ j\neq i}}^{k} \left(\frac{a}{p_j}\right) = (-1) \prod_{\substack{j=0 \\ j\neq i}}^{k} \left(\frac{1}{p_j}\right) = -1.$$

D'altra parte $a^{\frac{n-1}{2}} \not\equiv -1 \pmod{n}$. Infatti se fosse vero $a^{\frac{n-1}{2}} \equiv -1 \pmod{n}$ si avrebbe $a^{\frac{n-1}{2}} \equiv -1 \pmod{n/p_i}$, mentre invece da $a \equiv 1 \pmod{n/p_i}$ segue che $a^{\frac{n-1}{2}} \equiv 1 \pmod{n/p_i}$. Infine $\gcd(a, n) = 1$. Infatti se per assurdo $\gcd(a, n) > 1$ si avrebbe $p_i \mid a$ e quindi $\left(\frac{a}{p_i}\right) = 0$, contraddicendo la condizione $\left(\frac{a}{p_i}\right) = -1$. Per dimostrare l'esistenza del numero a soddisfacente le condizioni sopra citate, basta notare che tali condizioni sono equivalenti al sistema congruenze :

$$\begin{cases} a \equiv b \pmod{p_i} \\ a \equiv 1 \pmod{n/p_i} \end{cases}$$

dove b non è uno dei $\frac{p_i - 1}{2}$ resti non quadratici modulo p_i. Poichè $\gcd(p_i, n/p_i) = 1$, per il teorema cinese del resto, il sistema di congruenze ammette soluzione.

Sufficienza : $\exists a \in \mathbb{Z}_n^+ : \left(\frac{a}{n}\right) \not\equiv a^{\frac{n-1}{2}} \pmod{n}$ e $\gcd(a, n) = 1 \Rightarrow n \in \mathbb{N}\backslash\mathbb{P}$

Segue immediatamente dal criterio di Eulero. □

5.4 Probabilità di errore del Test di Solovay-Strassen

Definizione 48. *Sia $n \in \mathbb{N}\backslash\mathbb{P}$ e dispari, definiamo l'insieme dei* bugiardi di Eulero (alla primalità) di n *come :*

$$L_{SS}(n) = \{a \in \mathbb{Z}_n^* : Pr_{SS}(n, a) = false\}.$$

Teorema 61. *Sia $n \in \mathbb{N}\backslash\mathbb{P}$ e dispari, allora $\mid L_{SS}(n) \mid \leq \frac{\phi(n)}{2}$.*

Dimostrazione. Se n è un intero composto e dispari, sappiamo che esiste un $a \in \mathbb{Z}_n^*$ tale che $\left(\frac{a}{n}\right) \not\equiv a^{\frac{n-1}{2}} \pmod{n}$. Siano $b_1, b_2, ..., b_m$ i numeri positivi minori di n tali che $\gcd(b_j, n) = 1$ e $\left(\frac{b_j}{n}\right) \equiv b_j^{\frac{n-1}{2}} \pmod{n}$, per $j = 1, ..., m$ e siano $r_1, r_2, ..., r_m$ i resti positivi dei numeri $ab_1, ..., ab_m$ modulo n. Per il Teorema 29, notiamo che i numeri r_j formano un insieme di residui incongruenti modulo n e che $\gcd(r_j, n) = 1$ per ogni $j = 1, 2, ..., m$. Inoltre $\left(\frac{r_j}{n}\right) \not\equiv r_j^{\frac{n-1}{2}} \pmod{n}$ per ogni $j = 1, 2, ..., m$. Infatti se fosse vero $\left(\frac{r_j}{n}\right) \equiv r_j^{\frac{n-1}{2}} \pmod{n}$ per qualche j, avremmo che :

$$\left(\frac{ab_j}{n}\right) \equiv (ab_j)^{\frac{n-1}{2}} \pmod{n}$$

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

e poichè $\left(\frac{b_j}{n}\right) \equiv b_j^{\frac{n-1}{2}} \pmod{n}$ avremmo :

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

che contraddice l'ipotesi $\left(\frac{a}{n}\right) \not\equiv a^{\frac{n-1}{2}} \pmod{n}$. Quindi, notiamo che gli insiemi $\{b_1, ..., b_m\}$ e $\{r_1, ..., r_m\}$ sono insiemi disgiunti di numeri incongruenti modulo n, coprimi con n. Considerando che l'unione dei due insiemi, crea un insieme di $2m$ numeri coprimi con n e che il numero di coprimi con n è al più $\phi(n)$, si deve verificare che $2m \leq \phi(n)$. Da cui, otteniamo che $\mid L_{SS}(n) \mid = m \leq \frac{\phi(n)}{2}$. □

A questo punto, siamo in grado di determinare la probabilità che un numero composto e dispari n sia dichiarato primo dal test di Solovay-Strassen :

$$P[Pr_{SS}(n, a) = false] = \frac{\mid L_{SS}(n) \mid}{n-1} \leq \frac{\phi(n)}{2(n-1)} \leq \frac{1}{2}.$$

5.5 Complessità dell'algoritmo

La complessità dell'algoritmo di Solovay-Strassen richiede una preliminare analisi asintotica dell'algoritmo per il calcolo del simbolo di Jacobi. In particolare, analizzando la chiamata ricorsiva della funzione *jacobi* si nota una particolare analogia con la chiamata ricorsiva dell'algoritmo di Euclide per il calcolo del massimo comune divisore. Se rappresentiamo con $T_{gcd}(a, n)$ il numero di chiamate ricorsive dell'algoritmo di Euclide, calcolabile con la ricorrenza :

$$T_{gcd}(a,n) = \begin{cases} \mathcal{O}(1) & \text{se } n = 0 \\ T_{gcd}(n, a \bmod n) + \mathcal{O}(1) & \text{altrimenti} \end{cases}$$

sappiamo che tale numero è $\mathcal{O}(\log a)$ se $a < n$. Rappresentiamo con $T_J(a, n)$ il numero di chiamate ricorsive dell'algoritmo *jacobi*, attraverso la ricorrenza :

$$T_J(a,n) = \begin{cases} \mathcal{O}(1) & \text{se } a = 0 \\ T_J(n \bmod a', a') + \mathcal{O}(1) & \text{con } a = 2^{h'} \cdot a',\ h' \geq 0 \text{ e } a' \text{ dispari altrimenti.} \end{cases}$$

Poichè l'algoritmo di Solovay-Strassen richiama *jacobi* con $a < n$ e $n \bmod a' < a' \leq a$ si ha che $T_J(a, n) = \mathcal{O}(\log n)$. Inoltre, per ogni chiamata ricorsiva di *jacobi* si effettuano delle riduzioni modulari su quantità minori o uguali ad n. Pertanto la complessità di *jacobi* è $\mathcal{O}((\log n)^2)$. Possiamo a questo punto, determinare la complessità computazionale dell'algoritmo di Solovay-Strassen. Nel caso peggiore, esso richiama s volte la funzione *test-SS*. Per ogni chiamata della funzione *test-SS* si effettuano le seguenti operazioni:

1. si calcola $\gcd(a, n)$ (costo computazionale $\mathcal{O}((\log n)^3)$)
2. si determina il simbolo di Jacobi (costo computazionale $\mathcal{O}((\log n)^2)$)
3. si calcola un elevamento a potenza modulare (costo computazionale con l'algoritmo square-and-multiply $\mathcal{O}((\log n)^3)$)

per un totale di $\mathcal{O}(s((\log n)^2 + \mathcal{O}(2(\log n)^3))$ operazioni su bit. Concludiamo che l'algoritmo di Solovay-Strassen impiega $\mathcal{O}(s(\log n)^3)$.

Capitolo 6

L'algoritmo Agrawal-Kayal-Saxena

6.1 Il problema delle potenze perfette

Sia $n \geq 2$ un intero. In questa sezione, consideriamo il problema di stabilire se n è una potenza perfetta, cioè se esistono due interi $a \geq 2$ e $b \geq 2$ tali che $n = a^b$. Dal momento che b non può essere maggiore del $\log n$, questo problema si riduce a decidere se per un $2 \leq b \leq \log n$, esiste un intero $a \geq 2$ tale che $n = a^b$. Per un fissato $b \geq 2$, esponiamo un algoritmo che calcola l'intero $q = \lfloor n^{1/b} \rfloor$. Allora, n è un b-esima potenza perfetta se e solo se $q^b = n$. L'algoritmo che calcola questo $q = \lfloor n^{1/b} \rfloor$ è una versione, adattata ai numeri interi, dell'algoritmo Newton-Raphson ed è qui di seguito descritto:

Siano $n \geq 2$ e $b \geq 2$.

power(n, b)

1. $x_0 :=$ qualsiasi intero tale che $x_0^b \geq n$;
2. $i := 0$;
3. while $(x_i^b > n))$
4. $\quad x_{i+1} := \lfloor ((b-1)x_i + n/x_i^{b-1})/b \rfloor$
5. $\quad i := i + 1$;
6. return x_i;

6.1.1 Dimostrazione di Correttezza dell'algoritmo

Lemma 8. *Per ogni $i \geq 0$ per cui esiste x_i, si ha che $x_i \geq q = \lfloor n^{1/b} \rfloor$.*

Dimostrazione. Iniziamo col dimostrare che $x_0 \geq q$. Poichè l'algoritmo sceglie x_0 in modo tale che $x_0^b \geq n$, otteniamo che $x_0^b \geq n \geq \lfloor n^{1/b} \rfloor^b = q^b$ da cui segue $x_0 \geq q$. Consideriamo, adesso, la funzione

$$f(x) = (b-1)x^b - bn^{1/b}x^{b-1} + n.$$

Osserviamo che

$$f'(x) = b(b-1)x^{b-2}(x - n^{1/b}) > 0$$

per ogni $x > n^{1/b}$. Quindi, per ogni $x^b > n$, abbiamo che $f(x) > f(n^{1/b}) = 0$. Da $f(x) > 0$ otteniamo: $((b-1)x + n/x^{b-1})/b > n^{1/b}$. Quindi, se l'algoritmo va avanti per i iterazioni, $x_i^b > n$, ed esiste x_{i+1} tale che :

$$x_{i+1} = \lfloor ((b-1)x + n/x^{b-1})/b \rfloor \geq \lfloor n^{1/b} \rfloor = q.$$

□

Lemma 9. *Gli interi x_i, con $i \geq 0$ sono decrescenti.*

Dimostrazione. Da $x_i^b > n$ otteniamo che $x_i > \frac{n}{x_i^{b-1}}$. Quindi :

$$x_{i+1} = \lfloor ((b-1)x_i + n/x_i^{b-1})/b \rfloor \leq ((b-1)x_i + n/x_i^{b-1})/b < ((b-1)x_i + x_i)/b = x_i.$$

□

Teorema 62. *L'algoritmo **power**(n, b) calcola $\lfloor n^{1/b} \rfloor$.*

Dimostrazione. Poichè gli x_i sono decrescenti, il ciclo while terminerà sicuramente. L'output dell'algoritmo è il primo x_i tale che $x_i^b \leq n$. Poichè $x_i \leq n^{1/b}$ e x_i è un intero, noi abbiamo che $x_i \leq q$. Ma dal Lemma 8, $x_i \geq q$ e quindi l'output dell'algoritmo è q. □

6.1.2 Complessità dell'algoritmo

Osserviamo che l'algoritmo **power**(n, b) calcola $\lfloor n^{1/b} \rfloor$ per qualsiasi valore iniziale scelto per x_0 tale che $x_0 \geq n$. Per ottenere un buon limite superiore sul numero di iterazioni fatte dal ciclo while, noi dobbiamo scegliere un opportuno x_0.

Lemma 10. *Sia m un intero tale che $2^{(m-1)b} < n \leq 2^{mb}$, e sia $x_0 = 2^m$. Allora $x_0^b \geq n$ e $q \leq x_0 < 2n^{1/b}$.*

Dimostrazione. Notiamo che $x_0^b = 2^{mb} \geq n$ e

$$q = \lfloor n^{1/b} \rfloor \leq n^{1/b} \leq 2^m = x_0 = 2 \cdot 2^{m-1} < 2n^{1/b}.$$

□

Lemma 11. *Sia $i \geq 0$ tale che $n^{1/b} < x_i < 2n^{1/b}$. Allora x_{i+1} esiste, e*

$$x_{i+1} - n^{1/b} \leq \frac{b-1}{b+1}(x_i - n^{1/b}).$$

Dimostrazione. Consideriamo la funzione $g(x) = (b-1)x^b - 2bn^{1/b}x^{b-1} + (b+1)n$ e osserviamo che :

$$g'(x) = b(b-1)x^{b-2}(x - 2n^{1/b}) < 0$$

per ogni x tale che $n^{1/b} < x < 2n^{1/b}$. Dunque per ogni x tale che $n^{1/b} < x < 2n^{1/b}$ noi abbiamo che :

$$g(x) < g(n^{1/b}) = (b-1)n^{b\cdot 1/b} - 2bn^{1/b}(n^{1/b})^{b-1} + (b+1)n = (b-1)n - 2bn + (b+1)n = 0.$$

Ma da $g(x) = (b-1)x^b - 2bn^{1/b}x^{b-1} + (b+1)n < 0$, dividendo entrambi i membri per $x^{b-1}(b+1)$, otteniamo :

$$\frac{b-1}{b+1}x - \frac{2b}{b+1}n^{1/b} + \frac{n}{x^{b-1}} < 0$$

$$\frac{b^2-1-b^2+b}{b+1}x + \frac{-b^2-b+b^2-b}{b+1}n^{1/b} + \frac{n}{x^{b-1}} < 0$$

$$\left(b - 1 - b\frac{b-1}{b+1}\right)x + \left(-b + b\frac{b-1}{b+1}\right)n^{1/b} + \frac{n}{x^{b-1}} < 0$$

$$(b-1)x + \frac{n}{x^{b-1}} - bn^{1/b} < b\frac{b-1}{b+1}x - b\frac{b-1}{b+1}n^{1/b}$$

$$((b-1)x + \frac{n}{x^{b-1}})/b - n^{1/b} < \frac{b-1}{b+1}(x - n^{1/b}).$$

Poichè per ipotesi $n^{1/b} < x_i < 2n^{1/b}$ e inoltre

$$x_{i+1} - n^{1/b} \leq ((b-1)x_i + n/x_i^{b-1})/b - n^{1/b},$$

otteniamo

$$x_{i+1} - n^{1/b} \leq \frac{b-1}{b+1}(x_i - n^{1/b}).$$

□

Lemma 12. *Sia b un intero tale che $2 \leq b \leq \log n$, e sia x_0 definito come nel lemma 5. Il ciclo While dell'algoritmo **power**(n, b) richiede $\mathcal{O}(\log n)$ iterazioni.*

Dimostrazione. Denotiamo con $k+1$ il numero di iterazioni del ciclo while. E' chiaro che il ciclo While eseguirà il suo controllo almeno due volte, quindi $k \geq 1$. Dai Lemmi 9 e 10 segue che :

$$x_k \leq n^{1/b} < x_{k-1} < x_{k-2} < ... < x_0 < 2n^{1/b}.$$

Applicando il Lemma 11 ripetutamente, otteniamo che :

$$x_{k-1} - n^{1/b} \leq \left(\frac{b-1}{b+1}\right)^{k-1} (x_0 - n^{1/b}) < \left(\frac{b-1}{b+1}\right)^{k-1} n^{1/b},$$

e poichè $n^{1/b} < x_{k-1}$, abbiamo che $x_{k-1} - n^{1/b} \geq 1$. Segue che :

$$1 < \left(\frac{b-1}{b+1}\right)^{k-1} n^{1/b}.$$

Passando ai logaritmi :

$$0 < (k-1)\log\left(\frac{b-1}{b+1}\right) + \log n^{1/b}$$

e, quindi,

$$k - 1 < \frac{\log n^{1/b}}{\log\left(\frac{b+1}{b-1}\right)}.$$

Infine,

$$k < 1 + \frac{\log n^{1/b}}{\log((b+1)/(b-1))} \in \mathcal{O}\left(\frac{\log n}{b\log((b+1)/(b-1))}\right) = \mathcal{O}(\log n)$$

in quanto

$$\lim_{b\to\infty}\left(\frac{b+1}{b-1}\right)^b = \lim_{b\to\infty}\left(1+\frac{2}{b-1}\right)^{b-1}\left(1+\frac{2}{b-1}\right) = e^2$$

e quindi $\lim_{b\to\infty} b\log((b-1)/(b+1)) \to 2\log e$. Concludiamo che $k \in \mathcal{O}(\log n)$. □

Lemma 13. *Siano n, b degli interi, tali che $2 \leq b \leq \log n$. L'algoritmo **power**(n, b), con x_0 inizializzato come descritto dal Lemma 10, calcola $\lfloor n^{1/b} \rfloor$ in $\mathcal{O}((\log n)^3 \log\log n)$.*

Dimostrazione. Osserviamo dal Lemma 8 e 10 che $x_i^b \leq x_0^b < 2^b n \leq n^2$ per ogni i per il quale esiste x_i. Dunque, x_i^b e x_i^{b-1} possono essere calcolati in $\mathcal{O}((\log n)^2 \log\log n)$. Sappiamo già che il ciclo while esegue $\mathcal{O}(\log n)$ iterazioni, ognuna delle quali determina il valore x_{i+1}. Noti x_i^{b-1} e x_i^b, Il calcolo di x_{i+1} può essere fatto in $\mathcal{O}((\log n)^2)$. E quindi in totale la complessità è $\mathcal{O}((\log n)^3 \log\log n)$.

□

Teorema 63. *Sia $n \geq 2$ un intero. Esiste un algoritmo che stabilisce in tempo $\mathcal{O}((\log n)^4 \log\log n)$ se esistono o meno, due interi $a \geq 2$ e $b \geq 2$ tali che $n = a^b$.*

Dimostrazione. Se $n = a^b$ per interi $a \geq 2$ e $b \geq 2$, allora b deve essere minore o uguale a $\log n$. Dal Lemma 13, noi possiamo calcolare $q_b = \lfloor n^{1/b} \rfloor$ per ogni $2 \leq b \leq \log n$, in $\mathcal{O}((\log n)^4 \log\log n)$. Se $q_b^b = n$ per almeno un b allora n è della forma a^b. □

6.2 L'algoritmo di Agrawal, Kayal e Saxena

6.2.1 Idea dell'algoritmo Agrawal, Kayal e Saxena

L'idea dell'algoritmo nasce da una generalizzazione del piccolo teorema di Fermat :

Lemma 14. *Siano $a \in \mathbb{Z}$, $n \in \mathbb{N}$, con $n \geq 2$ e $gcd(a, n) = 1$. Allora n è primo se e solo se*

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

Dimostrazione. Per $0 < i < n$ il coefficiente di x^i in $(x + a)^n - (x^n + a)$ è $\binom{n}{i} a^{n-i}$.
Supponiamo che n sia primo. Allora $\binom{n}{i} \equiv 0 \pmod{n}$ per ogni $0 < i < n$ e quindi anche il polinomio $(x + a)^n - (x^n + a)$ è congruo zero modulo n.
Supponiamo che n sia composto. Consideriamo un divisore primo q di n e sia k tale che $q^k \mid n$ ma $q^{k+1} \nmid n$. Poichè $\gcd(a, n) = 1$ e $q^k \mid n$ si ha che $\gcd(a, q^k) = 1$ e quindi $\gcd(a^{n-q}, q^k) = 1$. Inoltre $q^k \nmid \binom{n}{q}$. Infatti, supponiamo per assurdo che $q^k \mid \binom{n}{q}$. Allora $\binom{n}{q} = \alpha q^k$ per qualche intero positivo α, cioè

$$\frac{n(n-1)(n-2)...(n-q+1)}{q!} = \alpha q^k$$

che si può riscrivere nella forma

$$n = \frac{\alpha(q-1)!q^{k+1}}{(n-1)(n-2)...(n-q+1)}.$$

Osserviamo che il secondo membro è un numero intero. Per ogni $1 \leq j \leq q-1$, poichè q è primo, $(n-j)$ è coprimo con q solo se $q \nmid (n-j)$. Ma $q \nmid (n-j)$, infatti se fosse $n-j \equiv 0 \pmod{q}$, sapendo che $n \equiv 0 \pmod{q}$ si avrebbe $j \equiv 0 \pmod{q}$, che è falso. Quindi $(n-j)$ è coprimo con q, da cui segue che $(n-j) \nmid q^{k+1}$ e che $(n-1)(n-2)...(n-q+1) \nmid q^{k+1}$ mentre

$$\frac{\alpha(q-1)!}{(n-1)(n-2)...(n-q+1)}$$

deve essere un intero. Questo significa che $q^{k+1} \mid n$. Assurdo.
Completiamo la dimostrazione, analizzando il coefficiente $\binom{n}{q}a^{n-q}$ di x^q nel polinomio $(x+a)^n - (x^n+a)$. Supponiamo per assurdo che tale coefficiente sia divisibile per n. Allora, esso a maggior ragione, sarebbe divisibile per q^k Ma poichè $\gcd(q^k, a^{n-q}) = 1$, si deve avere che $q^k \mid \binom{n}{q}$, assurdo. Pertanto il coefficiente di x^q non è congruo zero modulo n e quindi $(x+a)^n \not\equiv (x^n+a) \pmod{n}$. □

La precedente identità fornisce un semplice test per la primalità: "dato un numero naturale n, e scelto un a si controlla se la condizione $(x+a)^n \equiv (x^n+a) \pmod{n}$ è soddisfatta o meno". Questo controllo richiede però un tempo pari a $\Omega(n)$, cioè esponenziale rispetto al numero di cifre di n, perchè occorre valutare n coefficienti nel caso peggiore. Un modo semplice per ridurre il numero di coefficienti da valutare è quello di trasformare la condizione di primalità in :

$$(x+a)^n \equiv (x^n+a) \pmod{x^r-1, n}$$

per un r scelto opportunamente. E' chiaro che tutti i numeri primi soddisfano quest'ultima condizione, per qualsiasi valore di a e di r. Il problema è che anche qualche numero composto n potrebbe soddisfarla per certi valori di a ed r. Agrawal, Kayal e Saxena sono riusciti a dimostrare la seguente caratterizzazione : "Tutti i numeri composti che soddisfano la precedente congruenza non sono potenze di numeri primi. Inoltre, per tutti

i numeri composti, il numero di a e l'appropriato valore r che verificano la condizione di primalità, sono limitati superiormente dal $\log n$".

6.2.2 L'algoritmo

Sia $n \in \mathbb{N}$ e $n > 1$.

AKS(n)

1. if ($n = a^b$ per $a \in \mathbb{N}$ e $b > 1$) then return **composto**;
2. Trova il più piccolo r tale che $ord_r(n) > \log^2 n$;
3. if ($1 < \gcd(a, n) < n$ per qualche $a \leq r$) then return **composto**;
4. if ($n \leq r$) [1] then return **primo**;
5. for $a = 1$ to $\lfloor \sqrt{\phi(r)} \log n \rfloor$ do
6. if $((x + a)^n \not\equiv (x^n + a) \pmod{x^r - 1, n})$ then return **composto**;
7. return **primo**;

6.2.3 Dimostrazione di Correttezza dell'algoritmo

Lemma 15. *Se n è primo, allora l'algoritmo restituisce **primo***

Dimostrazione. Se n è un numero primo, l'algoritmo supera i controlli delle linee 1 e 3. Per il Lemma 14 il ciclo for della linea 5 non può mai ritornare **composto**, quindi n sarà dichiarato **primo** dalla linea 4 o dalla linea 6. □

Invertendo ipotesi e tesi del lemma precedente, si ottiene ancora una proposizione valida. Per dimostrare ciò, dobbiamo esporre una serie di risultati preliminari. Innanzitutto, se l'algoritmo restituisce **primo** nel passo 4 allora n deve essere primo altrimenti

[1] Il lemma 16 dimostra che questo passo è rilevante quando $n \leq \lceil \log^5 n \rceil$, cioè se e solo se $n \leq 5690034$

il passo 3 avrebbe trovato un fattore per n. Così l'unico caso che resta è quando l'algoritmo restituisce **primo** nel passo 7. Come notiamo, l'algoritmo presenta due cicli. Il primo ciclo determina un opportuno r tale che $ord_r(n) > \log^2 n$. Determiniamo, adesso, un limite superiore per r:

Lemma 16. *Esiste un* $r \leq \max\{3, \lceil \log^5 n \rceil\}$ *tale che* $ord_r(n) > \log^2 n$.

Dimostrazione. Se $n = 2$, il lemma è soddisfatto con $r = 3$. Infatti :

$$3 \leq \max\{3, 1\} = \max\{3, \lceil \log^5 2 \rceil\} \text{ e } ord_3(2) = 2 > 1 = \log^2 2.$$

Così assumiamo $n > 2$. Siano $r_1, r_2, ..., r_t$ tutti i numeri tali che $ord_{r_i}(n) \leq \log^2 n$ o tali che r_i divide n. Ciascuno di essi deve dividere il prodotto

$$P = n \cdot \prod_{i=1}^{\lfloor \log^2 n \rfloor} (n^i - 1) < n \cdot \prod_{i=1}^{\lfloor \log^2 n \rfloor} n^i = n^{1+\sum_{i=1}^{\lfloor \log^2 n \rfloor} i} =$$

$$= n^{\frac{1}{2}\lfloor \log^2 n \rfloor (\lfloor \log^2 n \rfloor + 1)+1} < n^{\frac{1}{2}\log^4 n + \frac{1}{2}\log^2 n + 1}.$$

Poichè $\log^2 n > \sqrt{2}$, si ha $\frac{1}{2}\log^2 n + 1 < \frac{1}{2}\log^4 n$ e quindi

$$P < n^{\log^4 n} = 2^{\log^5 n} \leq 2^{\lceil \log^5 n \rceil}.$$

Pertanto il $lcm\{r_1, r_2, ..., r_t\} < 2^{\lceil \log^5 n \rceil}$. Poichè $\lceil \log^5 n \rceil > 10$ per $n > 2$, dal Teorema 48, si ha $lcm\{1, 2, ..., \lceil \log^5 n \rceil\} \geq 2^{\lceil \log^5 n \rceil}$ e quindi $\{1, 2, ..., \lceil \log^5 n \rceil\} \not\subseteq \{r_1, ..., r_t\}$ cioè deve esistere un numero $s \leq \lceil \log^5 n \rceil$ tale che $s \notin \{r_1, r_2, ..., r_t\}$ e quindi $ord_s(n) > \log^2 n$. □

Aggiungiamo questo risultato :

Lemma 17. *Siano* $n, r \in \mathbb{N}$, *tali che* $ord_r(n) > 1$. *Allora esiste un divisore primo* p *di* n *tale che* $ord_r(p) > 1$.

Dimostrazione. Sia $n = p_1^{e_1} p_2^{e_2} ... p_k^{e_k}$ la fattorizzazione canonica di n. Supponiamo per assurdo che $ord_r(p_i) = 1$, per ogni $i = 1...k$. Questo significa che $p_i \equiv 1 \pmod{r} \forall i = 1...k$ e quindi

$$p_1^{e_1} \equiv 1 \pmod{r}$$

$$p_2^{e_2} \equiv 1 \pmod{r}$$

.....

$$p_r^{e_k} \equiv 1 \pmod{r}$$

da cui segue che $p_1^{e_1} p_2^{e_2} ... p_k^{e_k} \equiv 1 \pmod{r}$ cioè $n \equiv 1 \pmod{r}$. Assurdo perchè $ord_r(n) > 1$. □

Poichè l'algoritmo determina un r tale che $ord_r(n) > \log^2 n$ ed $n > 2$, sappiamo che $ord_r(n) > 1$ e per il lemma precedente n deve avere un divisore primo p tale che $ord_r(p) > 1$. Inoltre poichè $gcd(n, r) = 1$ (altrimenti l'algoritmo si sarebbe fermato al passo 3 o 4), si ha anche $p, n \in \mathbb{Z}_r^*$. Da questo momento, nel resto della trattazione p ed r resteranno fissati. Inoltre denoteremo con $s = \lfloor \sqrt{\phi(r)} \log n \rfloor$.

Il passo 5 dell'algoritmo verifica s equazioni. E ricordando l'ipotesi iniziale, ovvero che l'algoritmo restituisce **primo**, dal passo 6 ricaviamo

$$(x + a)^n \equiv x^n + a \pmod{x^r - 1, n} \ \forall a : 0 \leq a \leq s$$

Questo implica che :

$$(*)\ (x + a)^n \equiv x^n + a \pmod{x^r - 1, p} \ \forall a : 0 \leq a \leq s.$$

Dal Lemma 14, abbiamo

$$(**)\ (x + a)^p \equiv x^p + a \pmod{x^r - 1, p} \ \forall a : 0 \leq a \leq s.$$

Il nostro prossimo obiettivo è dimostrare che

$$(x + a)^{\frac{n}{p}} \equiv x^{\frac{n}{p}} + a \pmod{x^r - 1, p} \ \forall a : 0 \leq a \leq s.$$

A tal fine, dimostriamo i seguenti lemmi.

Lemma 18. *Per ogni $i \in \mathbb{N}$ si ha*

$$(x + a)^{n^i} \equiv x^{n^i} + a \pmod{x^r - 1, p}$$

Dimostrazione. Dimostriamo il lemma per induzione su i.

Caso base : Per $i = 0$ il lemma è banalmente vero.

Passo induttivo : Supponiamo vero il lemma per i_0. Dall'ipotesi induttiva,

$$(x+a)^{n^{i_0+1}} \equiv (x^{n^{i_0}} + a)^n \pmod{x^r - 1, p}$$

Sostituendo nella (*), $x^{n^{i_0}}$ al posto di x e osservando che $x^r - 1$ divide $x^{n^{i_0} \cdot r} - 1$ otteniamo

$$(x^{n^{i_0}} + a)^n \equiv x^{n^{i_0+1}} + a \pmod{x^r - 1, p}.$$

□

Definizione 49. *Dato un polinomio $f(x)$ e un numero $m \in \mathbb{N}$, diciamo che m è un* numero introspettivo *per $f(x)$ se*

$$[f(x)]^m \equiv f(x^m) \pmod{x^r - 1, p}.$$

Lemma 19. *Sia $q = n/p$. q è introspettivo per $x^p + a$.*

Dimostrazione. Dalla (**) abbiamo $(x^p + a)^q \equiv ((x+a)^p)^q \pmod{x^r - 1, p}$ e quindi :

$$(x^p + a)^q \equiv ((x+a)^p)^q \equiv (x+a)^{pq} \equiv x^{pq} + a \equiv (x^q)^p + a \pmod{x^r - 1, p}.$$

□

Lemma 20. *Sia $q = n/p$. q è introspettivo per $(x^p + a)^l$, per ogni $l \geq 1$.*

Dimostrazione.

$$((x^p + a)^l)^q \equiv ((x^p + a)^q)^l \equiv (x^{pq} + a)^l \equiv ((x^q)^p + a)^l \pmod{x^r - 1, p}.$$

□

Lemma 21. *Siano m, m_r interi tali che $m \equiv m_r \pmod{r}$. Allora, $x^m \equiv x^{m_r} \pmod{x^r - 1}$*

Dimostrazione. Sappiamo che $m = kr + m_r$ per qualche k intero. Quindi :

$$x^r \equiv 1 \pmod{x^r - 1}$$

$$x^{kr} \equiv 1 \pmod{x^r - 1}$$

$$x^{kr+m_r} \equiv x^{m_r} \pmod{x^r - 1}$$

$$x^m \equiv x^{m_r} \pmod{x^r - 1}.$$

□

Lemma 22. *Esiste un* $\bar{l}$ *tale che*

$$(x^p + a)^{\bar{l}} \equiv x + a \pmod{x^r - 1, p}$$

Dimostrazione. Sia $k = ord_r(n)$ (k esiste, perchè $\gcd(n, r) = 1$). Poniamo $\bar{l} = q \cdot n^{k-1}$. Per la (**) e il Lemma 18 si ha :

$$(x^p + a)^{\bar{l}} \equiv (x + a)^{pqn^{k-1}} \equiv (x + a)^{n^k} \equiv x^{n^k} + a \pmod{x^r - 1, p}.$$

Inoltre dal lemma precedente, poichè $n^k \equiv 1 \pmod{x^r - 1}$

$$x^{n^k} \equiv x \pmod{x^r - 1, p}.$$

Da cui segue :

$$(x^p + a)^{\bar{l}} \equiv x + a \pmod{x^r - 1, p}.$$

□

Lemma 23. *Se* $f(x) \equiv g(x) \pmod{x^r - 1, p}$ *e* q *è introspettivo per* $f(x)$ *allora* q *è introspettivo per* $g(x)$.

Dimostrazione. Per ipotesi $f(x) \equiv g(x) + h(x)(x^r - 1) \pmod{p}$ per qualche $h(x) \in \mathbb{Z}_p[x]$. Quindi

$$f(x^q) \equiv g(x^q) + h(x^q)(x^{qr} - 1) \pmod{p}$$

e poichè $x^r - 1$ divide $x^{qr} - 1$ si ha

$$f(x^q) \equiv g(x^q) \pmod{x^r - 1, p}.$$

Adesso,

$$(g(x))^q \equiv (f(x))^q \equiv f(x^q) \equiv g(x^q) \pmod{x^r - 1, p}.$$

□

Lemma 24.

$$(x+a)^{\frac{n}{p}} \equiv x^{\frac{n}{p}} + a \pmod{x^r - 1, p} \ \forall a : 1 \leq a \leq s.$$

Dimostrazione. Sia $q = n/p$, $\bar{l} = q \cdot n^{k-1}$, e $k = ord_r(n)$. Poichè q è introspettivo per $(x^p + a)^{\bar{l}}$ e $(x^p + a)^{\bar{l}} \equiv x + a \pmod{x^r - 1, p}$, si ha che q è anche introspettivo per $x + a$, da cui segue

$$(x+a)^{\frac{n}{p}} \equiv x^{\frac{n}{p}} + a \pmod{x^r - 1, p}.$$

□

Il seguente lemma mostra che i numeri introspettivi sono chiusi rispetto alla moltiplicazione:

Lemma 25. *Se m e m' sono numeri introspettivi per $f(x)$ allora anche $m \cdot m'$ è introspettivo per $f(x)$.*

Dimostrazione. Se m è introspettivo per $f(x)$ abbiamo che :

$$[f(x)]^m \equiv f(x^m) \pmod{x^r - 1, p}$$

da cui segue che :

$$[f(x)]^{m \cdot m'} \equiv [f(x^m)]^{m'} \pmod{x^r - 1, p}.$$

Ma anche m' è introspettivo per $f(x)$, quindi sostituendo x con x^m nell'equazione di introspezione per m' otteniamo che :

$$[f(x^m)]^{m'} \equiv f(x^{m \cdot m'}) \pmod{x^{m \cdot r} - 1, p}$$

$$[f(x^m)]^{m'} \equiv f(x^{m \cdot m'}) \pmod{x^r - 1, p} \text{ (poichè } x^r - 1 \text{ divide } x^{m \cdot r} - 1).$$

Uguagliando le congruenze precedenti otteniamo :

$$[f(x)]^{m \cdot m'} \equiv f(x^{m \cdot m'}) \pmod{x^r - 1, p}.$$

□

Per un dato m, l'insieme dei polinomi per i quali m è un numero introspettivo è chiuso rispetto alla moltiplicazione :

Lemma 26. *Se m è introspettivo per $f(x)$ e $g(x)$ allora esso è introspettivo per $f(x) \cdot g(x)$.*

Dimostrazione.

$$[f(x) \cdot g(x)]^m \equiv ([f(x)]^m \cdot [g(x)]^m) \equiv (f(x^m) \cdot g(x^m)) \pmod{x^r - 1, p}.$$

□

I due lemmi precedenti implicano che ogni numero dell'insieme $I = \{(\frac{n}{p})^i \cdot p^j \mid i, j \geq 0\}$ è introspettivo per qualsiasi polinomio dell'insieme $P = \{\prod_{a=0}^{s}(x+a)^{e_a} \mid e_a \geq 0\}$. Adesso, definiamo due gruppi basati su questi insiemi che giocano un ruolo cruciale per la dimostrazione.

Il primo gruppo è l'insieme dei residui modulo r dei numeri appartenenti ad I :

Definizione 50. $G = \{(\frac{n}{p})^i \cdot p^j \pmod{r} \mid i, j \geq 0\}$.

Questo è un sottogruppo di $\mathbb{Z}_r^*$, infatti da $\gcd(n, r) = \gcd(p, r) = 1$ segue che ogni elemento di I è coprimo con r. Sia $t =\mid G \mid$. G è generato da n e p modulo r e poichè $\mid G \mid \geq ord_r(n) > \log^2 n$, si ha che $t > \log^2 n$.
Il secondo gruppo è l'insieme di tutti i residui dei polinomi in P modulo $h(x)$ e p, dove $h(x)$ è un fattore irriducibile dell' r-esimo polinomio ciclotomico $Q_r(x)$ in $\mathbb{Z}_p[x]$.

Definizione 51. $H = \{\prod_{a=0}^{s}(x+a)^{e_a} \pmod{h(x), p} \mid e_a \geq 0\}$.

H è generato dagli elementi x, $x+1$, $x+2$, ..., $x+s$ in $\mathbb{Z}_p[x]/(h(x))$.

Lemma 27. $\mid H \mid \geq \binom{t+s}{t-1}$.

Dimostrazione. Iniziamo col dimostrare che dati due polinomi distinti $f(x)$ e $g(x)$ in P di grado minore a t, si ha

$$f(x) \not\equiv g(x) \pmod{h(x), p}.$$

Supponiamo per assurdo che $f(x) \equiv g(x) \pmod{h(x), p}$ e sia $m \in I$. Allora

$$f(x)^m \equiv g(x)^m \pmod{h(x), p}.$$

Ma m è introspettivo per entrambi i polinomi $f(x)$ e $g(x)$ e quindi :

$$f(x^m) \equiv f(x)^m \pmod{x^r - 1, p}$$

$$g(x^m) \equiv g(x)^m \pmod{x^r - 1, p}$$

e poichè $h(x)$ divide $x^r - 1$ otteniamo:

$$f(x^m) \equiv f(x)^m \pmod{h(x), p}$$

$$g(x^m) \equiv g(x)^m \pmod{h(x), p}$$

da cui segue che :

$$f(x^m) \equiv g(x^m) \pmod{h(x), p}.$$

Questo implica che x^m è una radice del polinomio $Q(y) = f(y) - g(y)$ per ogni $m \in G$. Pertanto i polinomi x^m con $m \in G$ sono t radici distinte di $Q(y)$. Assurdo, perchè $Q(y)$ ha grado minore di t per come $f(x)$ e $g(x)$ sono stati scelti. Quindi abbiamo dimostrato che polinomi distinti di grado minore a t in P mappano elementi distinti in $\mathbb{Z}_p[x]/(h(x))$.

Adesso, dimostriamo che $\sqrt{r} \log n < r$.

Poichè $n \in \mathbb{Z}_r^*$ abbiamo che $n^{\phi(r)} \equiv 1 \pmod{r}$ e quindi $ord_r(n)$ deve dividere $\phi(r)$. Dal Lemma 16, segue :

$$\log^2 n < ord_r(n) < \phi(r) < r$$

$$\log n < \sqrt{r}$$

$$\sqrt{r} \log n < r.$$

Osserviamo, dunque, che per ogni i, j tali che $1 \leq i \neq j \leq s = \lfloor \sqrt{\phi(r)} \log n \rfloor < \sqrt{r} \log n < r < p$ i polinomi $x, x+1, x+2, ..., x+s$ sono tutti distinti in G e quindi anche in $\mathbb{Z}_p[x]/(h(x))$. Inoltre $deg(h) = ord_r(p) > 1$, e

$$x + a \not\equiv 0 \pmod{h(x), p} \text{ per ogni } 0 \leq a \leq s.$$

Così devono esistere $s+1$ polinomi distinti di grado uno in H e con questi polinomi, possiamo costruire $\sum_{i=0}^{t-1} \binom{s+1+i-1}{i}$ distinti polinomi di grado minore a t in H. Per completare la dimostrazione è sufficiente dimostrare che

$$\sum_{i=0}^{t-1} \binom{s+i}{i} \geq \binom{t+s}{t-1}.$$

Ciò può essere fatto per induzione su t.
Caso base : se $t = 1$ allora $\binom{s}{0} \geq \binom{s+1}{0}$

Passo induttivo : Supponiamo vera la tesi per $t_0 - 1$ e dimostriamola per t_0

$$\sum_{i=0}^{t_0} \binom{s+i}{i} = \sum_{i=0}^{t_0-1} \binom{s+i}{i} + \binom{s+t_0}{t_0} \geq \binom{t_0+s}{t_0-1} + \binom{t_0+s}{t_0} = \binom{t_0+1+s}{t_0}.$$

□

Nel caso in cui n non è una potenza di p la dimensione di H può essere limitata superiormente.

Lemma 28. *Se n non è una potenza di p allora $\mid H \mid \leq n^{\sqrt{t}}$.*

Dimostrazione. Consideriamo il seguente sottoinsieme di I:

$$\hat{I} = \{(\frac{n}{p})^i \cdot p^j \mid 0 \leq i, j \leq \lfloor \sqrt{t} \rfloor\}.$$

Se n non è un potenza di p allora l'insieme $\hat{I}$ ha $(\lfloor \sqrt{t} \rfloor + 1)^2$ numeri distinti. E poichè $(\lfloor \sqrt{t} \rfloor + 1)^2 = (\lfloor \sqrt{t} + 1 \rfloor)^2 > t$ e $\mid G \mid = t$, devono esistere almeno due numeri congrui modulo r in $\hat{I}$. Siano essi m_1 e m_2 con $m_1 > m_2$. Così noi abbiamo che

$$x^{m_1} \equiv x^{m_2} \pmod{x^r - 1}.$$

Sia $f(x) \in P$. Allora,

$$[f(x)]^{m_1} \equiv f(x^{m_1}) \equiv f(x^{m_2}) \equiv [f(x)]^{m_2} \pmod{x^r - 1, p}.$$

E poichè $h(x)$ divide $x^r - 1$,

$$[f(x)]^{m_1} \equiv [f(x)]^{m_2} \pmod{h(x), p}.$$

Quindi, ciascun polinomio $f(x) \in H$ è una radice[2] del polinomio $Q'(y) = y^{m_1} - y^{m_2}$ in $\mathbb{Z}_p[x]/(h(x))$. Così possiamo affermare che $Q'(y)$ ha almeno $\mid H \mid$ radici distinte in $\mathbb{Z}_p[x]/(h(x))$. Ma il grado di $Q'(y)$ è $m_1 \leq (\frac{n}{p} \cdot p)^{\lfloor\sqrt{t}\rfloor} \leq n^{\sqrt{t}}$. Questo mostra che $\mid H \mid \leq n^{\sqrt{t}}$. □

Lemma 29. *Se l'algoritmo restituisce **primo**, allora n è primo.*

Dimostrazione. Abbiamo già osservato che se l'algoritmo restituisce **primo** alla linea 4, allora n è primo, perchè altrimenti la linea 3 troverebbe un fattore diverso da 1 e n, che divide n.

Armati dei risultati ottenuti, e supponendo che l'algoritmo restituisca **primo** alla linea 7, sappiamo che

$$\mid H \mid \geq \binom{s+t}{t-1} = \binom{s+t}{s+1}.$$

Da $ord_r(n) = t > \log^2 n$ otteniamo

$$\sqrt{t} > \log n$$

$$t > \sqrt{t} \log n$$

e poichè t è un intero si ha $t \geq \lfloor\sqrt{t} \log n\rfloor + 1$.

Dal Corollario 7, segue:

$$\mid H \mid \geq \binom{s+t}{s+1} \geq \binom{s+1+\lfloor\sqrt{t}\log n\rfloor}{s+1} = \binom{s+1+\lfloor\sqrt{t}\log n\rfloor}{\lfloor\sqrt{t}\log n\rfloor}.$$

[2]Questo risultato è il prodotto di una comunicazione privata tra gli autori dell'algoritmo e Adam Kalai, Amit Sahai e Madhu Sudan

Ma G è sottogruppo di $\mathbb{Z}_r^*$ e quindi $t =\mid G \mid \leq \mid \mathbb{Z}_r^* \mid = \phi(r)$. Otteniamo, immediatamente, che $s = \lfloor \sqrt{\phi(r)} \log n \rfloor \geq \lfloor \sqrt{t} \log n \rfloor$ e dal Corollario 7 :

$$\mid H \mid \geq \binom{s + 1 + \lfloor \sqrt{t} \log n \rfloor}{\lfloor \sqrt{t} \log n \rfloor} \geq \binom{2\lfloor \sqrt{t} \log n \rfloor + 1}{\lfloor \sqrt{t} \log n \rfloor}.$$

Da $t > \log^2 n$ segue

$$\sqrt{t} > \log n \Rightarrow \sqrt{t} \log n > \log^2 n \Rightarrow \lfloor \sqrt{t} \log n \rfloor > \lfloor \log n \rfloor > 1$$

e applicando il Teorema 46 abbiamo :

$$\mid H \mid \geq \binom{2\lfloor \sqrt{t} \log n \rfloor + 1}{\lfloor \sqrt{t} \log n \rfloor} > 2^{\lfloor \sqrt{t} \log n \rfloor + 1} = 2^{\lceil \sqrt{t} \log n \rceil} \geq 2^{\sqrt{t} \log n} = n^{\sqrt{t}}.$$

Abbiamo già dimostrato che se n non è una potenza di p allora $\mid H \mid \leq n^{\sqrt{t}}$, ma $\mid H \mid \geq n^{\sqrt{t}}$, e quindi $n = p^k$ per qualche $k > 0$. Se $k > 1$, l'algoritmo restituisce **composto** alla prima linea. Concludiamo che $n = p$. □

Teorema 64. *L'algoritmo restituisce **primo** se e solo se n è primo.*

Dimostrazione. Segue dal lemma precedente e dal Lemma 10. □

6.2.4 Complessità dell'algoritmo

Il tempo dominante nell' esecuzione dell' algoritmo AKS, è associato alla linea 5 e 6, dove per ogni valore a del ciclo for si devono calcolare i coefficienti dei polinomi $(x+a)^n \mod (x^r - 1)$ e $x^n + a \mod (x^r - 1)$ in $\mathbb{Z}_n[x]$ e confrontarli. Esponiamo adesso una variante, adattata ai polinomi, dell'algoritmo square and multiply. L'algorimo calcola $(x + a)^n \mod (x^r - 1)$, ma è chiaro che si può applicare anche al calcolo di $x^n + a \mod (x^r - 1)$:

Siano n, r, a interi con $2 \leq r < n$ e $1 \leq a < n$.

1. $f(x) := 1$; $g(x) := x + a$; $y := n$;

2. while $(y \neq 0)$

3. if (y è pari)

4. $y := y/2$

5. $h(x) := g(x)g(x)$

6. $g(x) := h(x) \pmod{x^r - 1}$;

7. else

8. $y := y - 1$

9. $h(x) := f(x)g(x)$

10. $f(x) := h(x) \pmod{x^r - 1}$;

11. return $f(x)$;

Assumendo che nelle operazioni in cui sono coinvolti $f(x), g(x)$ e $h(x)$, i coefficienti sono ridotti modulo n, l'algoritmo restituisce la quantità

$$f(x)^{(y \bmod 2)} g(x)^y \equiv (x + a)^n \pmod{x^r - 1} \text{ in } \mathbb{Z}_n[x].$$

Lemma 30. *Siano n, r, a interi con $2 \leq r < n$ e $1 \leq a < n$. I coefficienti del polinomio $(x + a)^n \mod (x^r - 1)$ in $\mathbb{Z}_n[x]$ possono essere calcolati in tempo $\mathcal{O}((\log n)^3 \cdot r \log r)$.*

Dimostrazione. Innanzitutto, osserviamo che i gradi dei polinomi $f(x)$ e $g(x)$ sono sempre minori o uguali a $r - 1$. Quindi il grado di $h(x)$ è sempre minore o uguale a $2r - 2$. L'assegnamento $h(x) := g(x)g(x)$ richiede un tempo pari a $\mathcal{O}(r \log r)$, se si usa il metodo della Trasformata veloce di Fourier. Inoltre, ciascun coefficiente ottenuto dalla moltiplicazione deve essere ridotto modulo n. Quindi $h(x)$ può essere calcolato in tempo $\mathcal{O}((\log n)^2 \cdot r \log r)$. Lo stesso tempo è richiesto per l'assegnamento $h(x) := f(x)g(x)$. Il polinomio $h(x)$ può essere scritto come $h(x) = \sum_{i=0}^{2r-2} h_i x^i$, dove ciascun coefficiente h_i è un intero appartenente all'intervallo $[0, n - 1]$. Allora,

$$h(x) \mod (x^r - 1) = \sum_{i=0}^{r-2}((h_i + h_{r+i}) \mod n)x^i + (h_{r-1} \mod n)x^{r-1}.$$

Quindi, la riduzione $h(x) \mod (x^r - 1)$ richiede un tempo pari a $\mathcal{O}(r(\log n)^2)$ e dal momento che l'algoritmo esegue $\mathcal{O}(\log n)$ iterazioni (ciclo while), segue la tesi. □

Teorema 65. *La complessità asintotica dell'algoritmo $AKS(n)$ è $\mathcal{O}((\log n)^{23/2} \log\log n)$*

Dimostrazione. Analizziamo la complessità asintotica di ciascuna linea.
La linea 1 richiede un tempo pari a $\mathcal{O}((\log n)^4 \cdot \log\log n)$.
Nella linea 2, si trova un r tale che $ord_r(n) > (\log n)^2$. Ciò si può fare, provando per valori incrementali di r la condizione $n^k \equiv 1 \pmod r$ per ogni $k \leq (\log n)^2$. Per un particolare r, provare questa condizione richiede $\mathcal{O}((\log n)^2)$ moltiplicazioni modulo r per un totale di $\mathcal{O}((\log n)^2(\log r)^2)$. E considerando che $r = \mathcal{O}((\log n)^5)$ la linea 2 si esegue in $\mathcal{O}((\log n)^2 \cdot (\log(\log n)^5)^2) = \mathcal{O}((\log n)^2(\log\log n)^2)$.
La linea 3 implica il calcolo del massimo comune divisore dei primi r numeri con n. Poichè $a \leq r < n$, ciascun calcolo di $\gcd(a, n)$ impiega $\mathcal{O}((\log n)^3)$ con l'algoritmo di Euclide. E considerando che $r = \mathcal{O}((\log n)^5)$, la linea 3 si esegue in $\mathcal{O}(r \cdot (\log n)^3) = \mathcal{O}((\log n)^8)$.
La linea 4 richiede appena $\mathcal{O}(\log n)$.
La linea 5 prevede la verifica di $\lfloor \sqrt{\phi(r)} \log n \rfloor$ equazioni. Ciascuna equazione può essere verificata in tempo $\mathcal{O}((\log n)^3 \cdot r \log r)$. Per un totale di operazioni pari a

$$\mathcal{O}((\log n)^4 \cdot r\sqrt{\phi(r)} \log r) = \mathcal{O}((\log n)^4 \cdot r\sqrt{r} \log r) = \mathcal{O}((\log n)^4 \cdot r^{3/2} \log r) =$$

$$= \mathcal{O}((\log n)^4 \cdot (\log n)^{15/2} \log((\log n)^5)) = \mathcal{O}((\log n)^{23/2} \log\log n).$$

Chiaramente, la complessità dominante nell'algoritmo è data dalla linea 5, e quindi segue la tesi. □

Appendice A

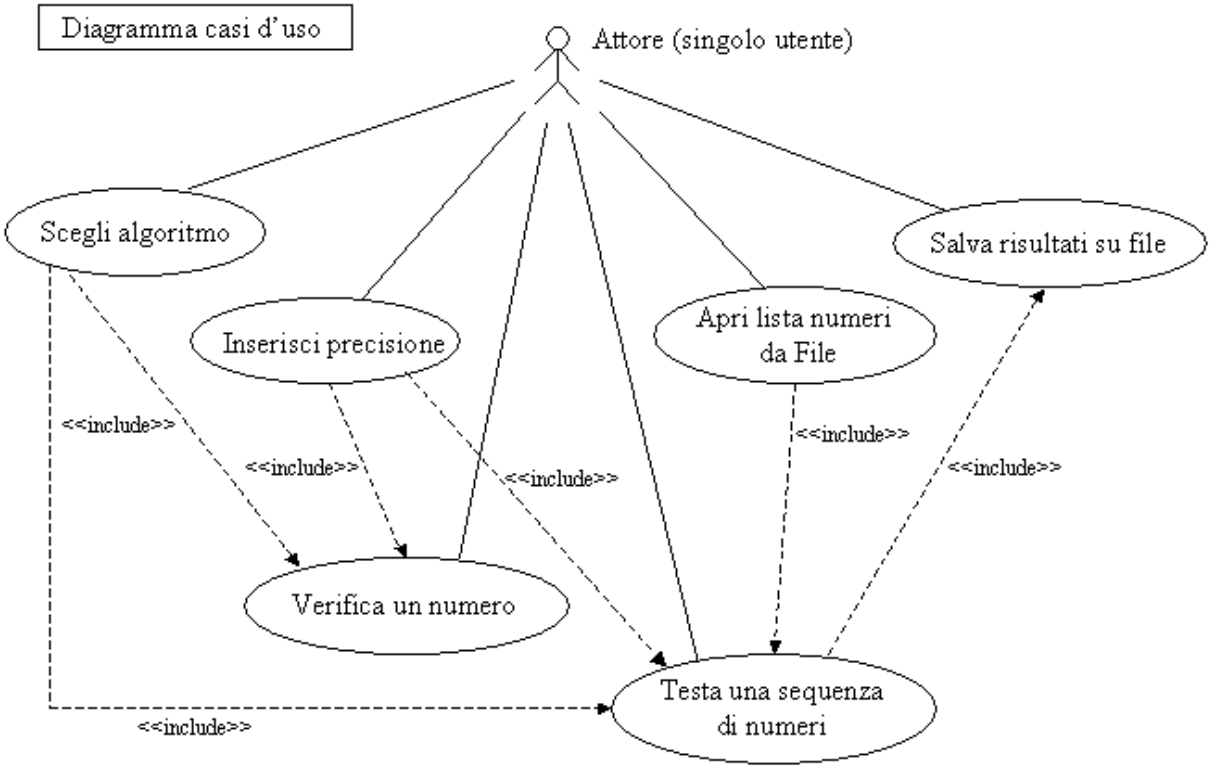
L'Implementazione

A.1 Introduzione

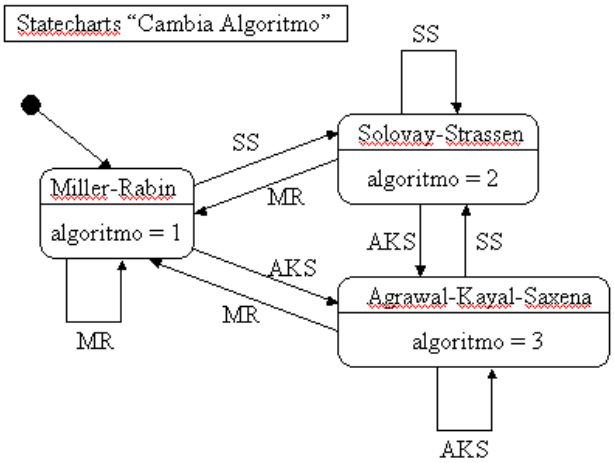
In questa sezione verrà descritta tutta la fase di progettazione del software allegato alla tesi. Chiaramente il processo di traduzione degli algoritmi in un vero e proprio software, può essere visto come un semplice lavoro di programmazione, viste le ridotte dimensioni del programma (1080 linee di codice). Ma l'esigenza di manutenibilità legata alla continua evoluzione dell'algoritmo AKS e la difficoltà nel calcolo di certe funzioni (assenti nelle librerie a precisione illimitata) richiedono un processo di ingegnerizzazione e di descrizione delle soluzioni adottate. Facendo uso di appropriati diagrammi UML, ho suddiviso questa sezione in tre parti : Analisi dei requisiti, Progettazione e Validazione.

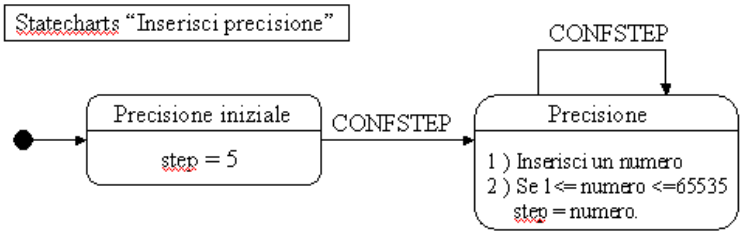
A.2 Analisi dei Requisiti

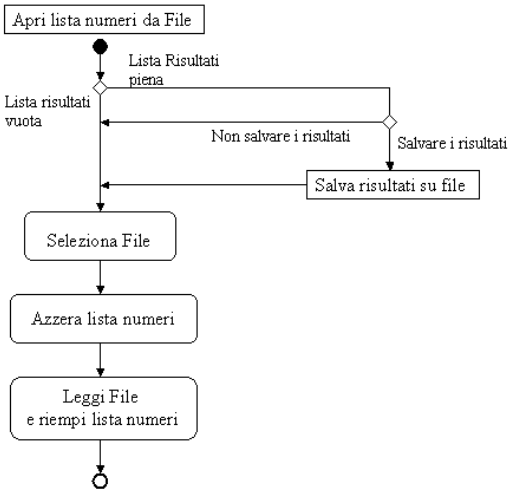
L'analisi dei Requisiti è il processo che definisce i servizi e i vincoli del software che dovrà essere prodotto. Nel caso specifico, l'obiettivo è implementare gli algoritmi Miller-Rabin, Solovay-Strassen e Agrawal-Kayal-Saxena. Si è ritenuto di non implementare il crivello di Eratostene, per la sua complessità esponenziale rispetto alla dimensione dell'input. E' richiesta una interfaccia grafica che consente all'utente di selezionare un algoritmo e di testare, in funzione dell'algoritmo scelto, la primalità di un numero inserito da tastiera o di una lista di numeri presente in un file di testo. Quando si controlla una lista di numeri, l'utente deve poter memorizzare i risultati (primalità e tempo di elaborazione per ogni numero) su un altrettanto file di testo. Il seguente diagramma mostra le attività che potrà svolgere l'utente (pagina successiva):

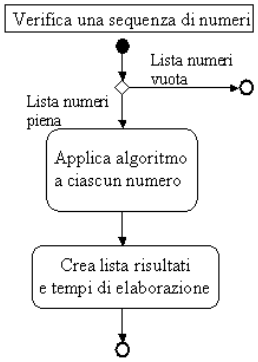


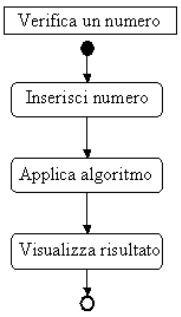
Ciascuna di queste attività può essere descritta con un apposito diagramma di attività. Per le attività "Scegli Algoritmo "e "Inserisci Precisione "utilizziamo dei diagrammi di stato, anche se manteniamo ancora indefinita la modalità di generazione degli eventi. Il primo diagramma di stato mostra che il software userà, per default, l'algoritmo Miller-Rabin ma di fronte al verificarsi di certi eventi (MR, SS, AKS), il programma potrà modificare il suo stato impostando volta per volta l'algoritmo corrente. Il secondo diagramma di stato invece mostra come impostare la precisione degli algoritmi probabilistici. Per default la precisione è 5. Gli altri diagrammi di attività sono piuttosto intuitivi.

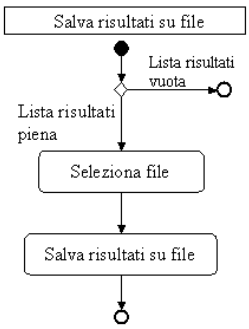

Statecharts "Cambia Algoritmo"
SS
Solovay-Strassen
algoritmo = 2
SS
MR
Miller-Rabin
algoritmo = 1
AKS
SS
AKS
MR
MR
Agrawal-Kayal-Saxena
algoritmo = 3
AKS


Statecharts "Inserisci precisione"
CONFSTEP
Precisione iniziale
step = 5
CONFSTEP
Precisione
1) Inserisci un numero
2) Se 1<= numero <=65535
step = numero.


Apri lista numeri da File
Lista Risultati piena
Lista risultati vuota
Non salvare i risultati
Salvare i risultati
Salva risultati su file
Seleziona File
Azzera lista numeri
Leggi File e riempi lista numeri


Verifica una sequenza di numeri
Lista numeri vuota
Lista numeri piena
Applica algoritmo a ciascun numero
Crea lista risultati e tempi di elaborazione


Verifica un numero
Inserisci numero
Applica algoritmo
Visualizza risultato


Salva risultati su file
Lista risultati vuota
Lista risultati piena
Seleziona file
Salva risultati su file

A.3 Progettazione

Dopo aver descritto le funzionalità che il software deve fornire all'utente, vediamo come queste funzionalità sono state realizzate. Innanzitutto, è necessario uno studio di fattibilità, cioè stabilire se è possibile o meno fornire le funzionalità richieste con gli strumenti hardware e software disponibili. Diciamo che i vincoli imposti sulla scelta del linguaggio sono:

1. Presenza di una libreria matematica a precisione illimitata
2. Possibilità di sviluppare una interfaccia grafica.

La scelta è caduta sul linguaggio Java, che oltre ad essere multipiattaforma, fornisce già le librerie AWT/Swing per le interfaccie grafiche e le classi BigInteger e BigDecimal per le elaborazioni sui grandi numeri. In relazione alle classi BigInteger e BigDecimal, è importante notare l'assenza di un metodo per il calcolo del logaritmo in base due (utile per l'algoritmo AKS). La classe BigInteger fornisce un metodo bitLength() che applicato ad un numero n restituisce $\lfloor \log(n+1) \rfloor$. Questo valore può essere talmente approssimato da alterare il risultato finale dell'algoritmo, anche su piccoli numeri. La soluzione adottata consente di calcolare il logaritmo approssimato al millesimo, anche se introduce un ritardo di elaborazione, e non può ritenersi una soluzione adeguata per tutti i numeri (l'approssimazione andrebbe adattata alla grandezza del numero). L'idea applicata è questa:
Poichè

$$\lfloor \log n \rfloor \leq \log n \leq \lfloor \log n \rfloor + 1$$

si ha

$$2^{\lfloor \log n \rfloor} \leq n \leq 2^{\lfloor \log n \rfloor} \cdot 2$$

Quindi possiamo impostare un ciclo for nella variabile j da 0 a 1000 che confronta tutte le quantità $2^{\lfloor \log n \rfloor} \cdot 2^{\frac{1}{1000} \cdot j}$ con n alla ricerca di quel valore che risulta essere maggiore di n. A quel punto, il logaritmo di n approssimato per difetto al millesimo sarà $\lfloor \log n \rfloor + \frac{1}{1000} \cdot (j-1)$. Tenendo presente che $2^{\frac{1}{1000}} = 1.000693387$ e che tutte le operazioni di questo

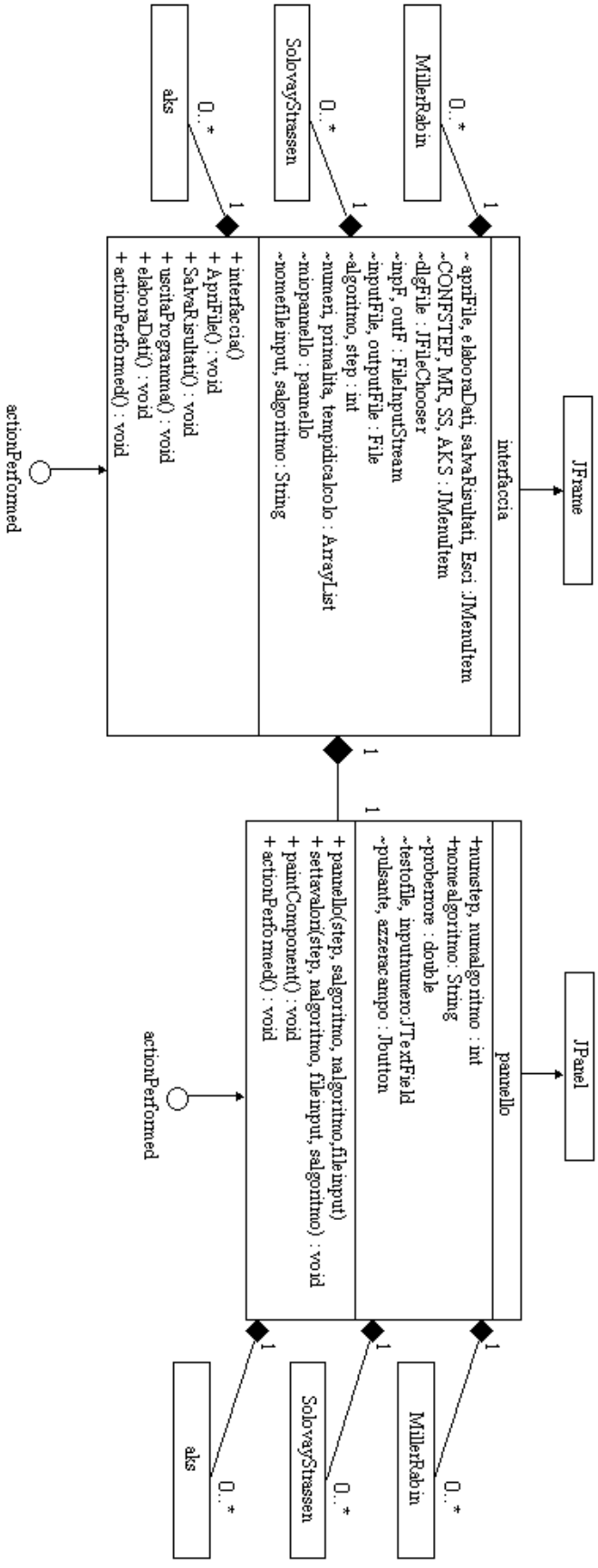
algoritmo possono essere espresse con metodi di BigInteger e BigDecimal, anche questo vincolo risulta superato. Un analogo problema di approssimazione riguarda il calcolo della radice quadrata di un numero. Le classi BigInteger e BigDecimal, non offrono al programmatore un metodo per ottenere la radice quadrata, ma forniscono un metodo per l'elevamento a potenza pow(). Una soluzione potrebbe essere:

1. il calcolo del $\lfloor\sqrt{n}\rfloor$, con un ciclo nella variabile i che a partire da 1 e con incrementi unitari controlla se $i^2 > n$. Quando si verifica la condizione si dichiara $\lfloor\sqrt{n}\rfloor = i-1$

2. Approssimare la radice quadrata al millesimo con un ciclo che aggiunge volta per volta alla base (inizialmente pari a $\lfloor\sqrt{n}\rfloor$) il numero 0.001 e che controlla se il quadrato di questa base è maggiore di n.

Infine, l'ultimo vincolo da superare è il controllo delle congruenze $(x + a)^n \not\equiv (x^n + a) \pmod{x^r - 1, n}$. Cercando di riprodurre fedelmente la teoria si può costruire una libreria polinomiale che permetta la moltiplicazione di polinomi con il metodo basato sulla FFT. Ma le differenze computazionali tra il metodo di moltiplicazione tradizionale e il metodo di moltiplicazione basato sulla FFT sono trascurabili, e non alterano la classe computazionale di appartenenza dell'algoritmo, che resta sempre polinomiale. Quindi, per semplicità, si può optare per una libreria polinomiale già pronta trascurando i dettagli di complessità. E' stata scelta, per ragioni di velocità e di facilità d'uso, la libreria perisic disponibile all'indirizzo web http://ring.perisic.com/.

La seconda fase della progettazione, consiste nell'identificazione delle classi da implementare. Questa operazione è strettamente legata all'analisi grammaticale dei requisiti. Il testo del problema già fornisce delle indicazioni : E' necessario costruire una classe per ogni tipo di algoritmo e specializzare un JFrame e un JPanel per l'interfaccia grafica. Concentriamo la nostra attenzione sull'interfaccia grafica.


JFrame
interfaccia
~ apriFile, elaboraDati, salvaRisultati, Esci :JMenuItem
~CONFSTEP, MR, SS, AKS : JMenuItem
~dlgFile : JFileChooser
~inpF, outF : FileInputStream
~inputFile, outputFile : File
~algoritmo, step : int
~numeri, primalita, tempidicalcolo : ArrayList
~miopannello : pannello
~nomefileinput, salgoritmo : String
+ interfaccia()
+ ApriFile() : void
+ SalvaRisultati() : void
+ uscitaProgramma() : void
+ elaboraDati() : void
+ actionPerformed() : void
actionPerformed
MillerRabin
SolovayStrassen
aks
0..*
1
JPanel
pannello
+numstep, numalgoritmo : int
+nomealgoritmo: String
~probcorrette : double
~testofile, inputnumero:JTextField
~pulsante, azzeracampo : Jbutton
+ pannello(step, salgoritmo, nalgoritmo,fileinput)
+ settavaloni(step, nalgoritmo, fileinput, salgoritmo) : void
+ paintComponent() :void
+ actionPerformed() : void
actionPerformed

Come notiamo dal diagramma UML, la classe *interfaccia* definisce :

1. un menù a tendina attraverso i bottoni JMenuItem, che ha lo scopo di generare gli eventi (vedi Diagrammi di stato) e di avviare le attività dell'utente (vedi Diagrammi delle attività)

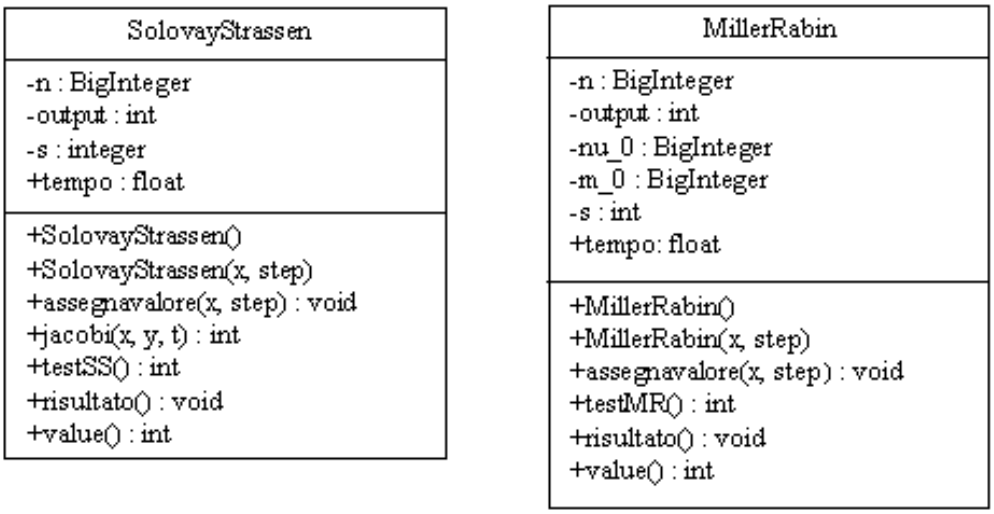
2. delle variabili (*algoritmo, step, ecc.*) che memorizzano lo stato corrente del software

3. Tre liste : *numeri, primalita e tempidicalcolo*, della stessa dimensione e che contengono i dati e i risultati delle elaborazioni

4. un pannello personalizzato : *miopannello.*

L'interfaccia *actionPerformed()* controlla quale evento è stato generato dall'utente e in funzione dell'evento, richiama i metodi : *Aprifile(), SalvaRisultati(), uscitaProgramma(), elaboraDati().* I primi due metodi implementano rispettivamente le attività "Apri lista numeri da File"e "Salva risultati su File". *uscitaProgramma()* è stato aggiunto per gestire le operazioni di chiusura della finestra. *elaboraDati()* seleziona ciascun numero della lista *numeri*, e lo passa come parametro ai metodi *settavalori(..)* o *assegnavalore(..)* dell'oggetto corrispondente all'algoritmo scelto, per testarne la primalità. Il risultato e il tempo di elaborazione di ogni controllo verranno successivamente inseriti nelle rispettive posizioni degli array *primalita* e *tempidielaborazione.* Il pannello *miopannello* contiene solo oggetti grafici e ha il compito di visualizzare all'utente, attraverso *paintComponent()*, le impostazioni correnti del software. Inoltre l'interfaccia *actionPerformed()*, alla pressione del JButton *pulsante* estrae il numero del JTextField *inputnumero* e ne verifica la primalità.

Descriviamo, adesso, le tre classi relative agli algoritmi. Le classi *MillerRabin* e *SolovayStrassen* sono molto simili :



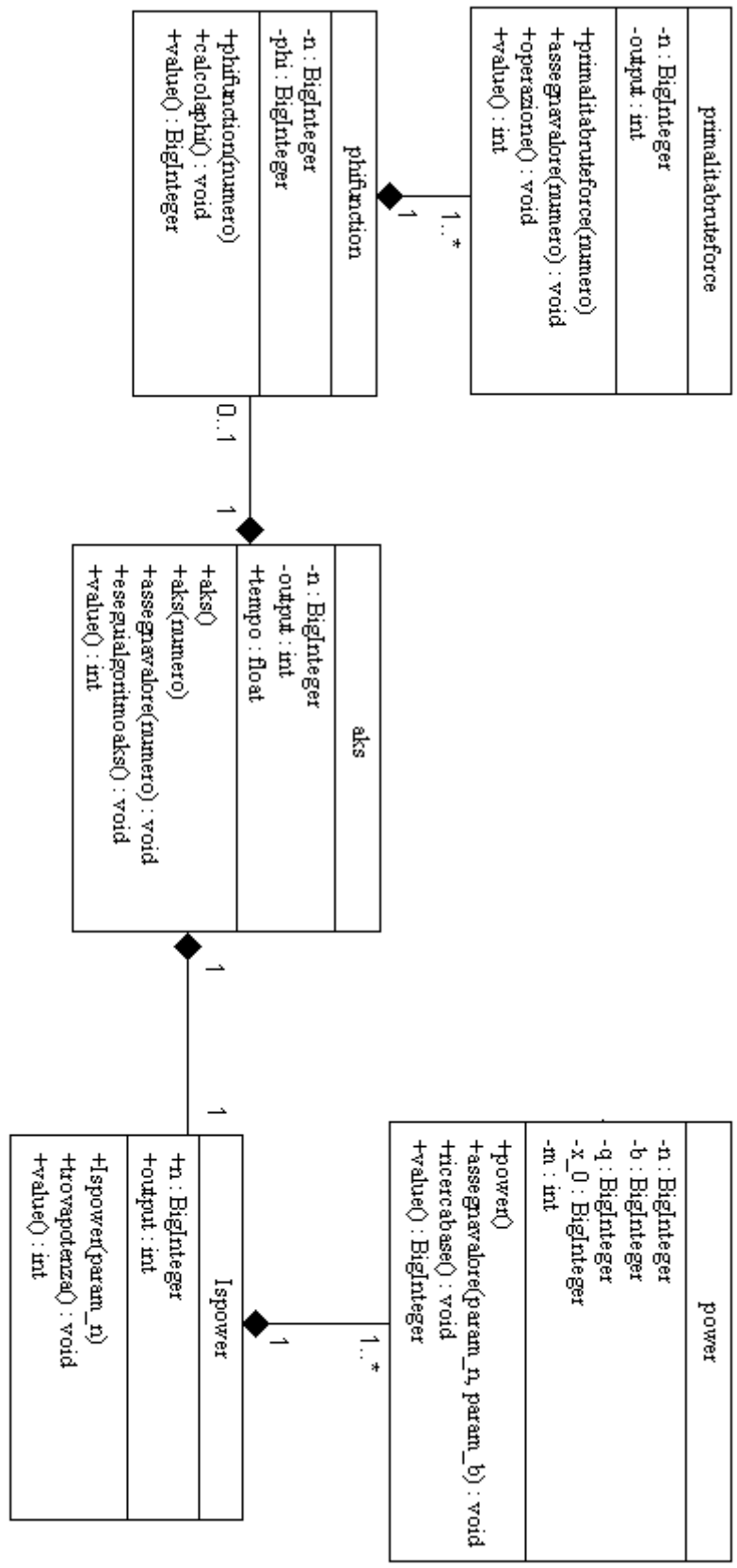
Entrambe definiscono come campi :

1. un BigInteger n (il numero da verificare)
2. l'intero *output* che conterrà alla fine delle operazioni 1 se n è primo, o 0 se n è composto
3. un intero s che specifica il numero di volte che il metodo *testSS()* o *testMR()* dovrà essere ripetuto dal metodo *risultato()*.
4. la variabile *tempo*, che conterrà il tempo di elaborazione.

Entrambe le classi hanno un metodo *assegnavalore(..)*, che inizializza n ed s, un metodo *value()* che restituisce il valore di *output* e un metodo *risultato()* che avvia il test probabilistico corrispondente (*testMR()* o *testSS()*). Le uniche differenze, oltre all'implementazione di *testMR()* e *testSS()*, sono :

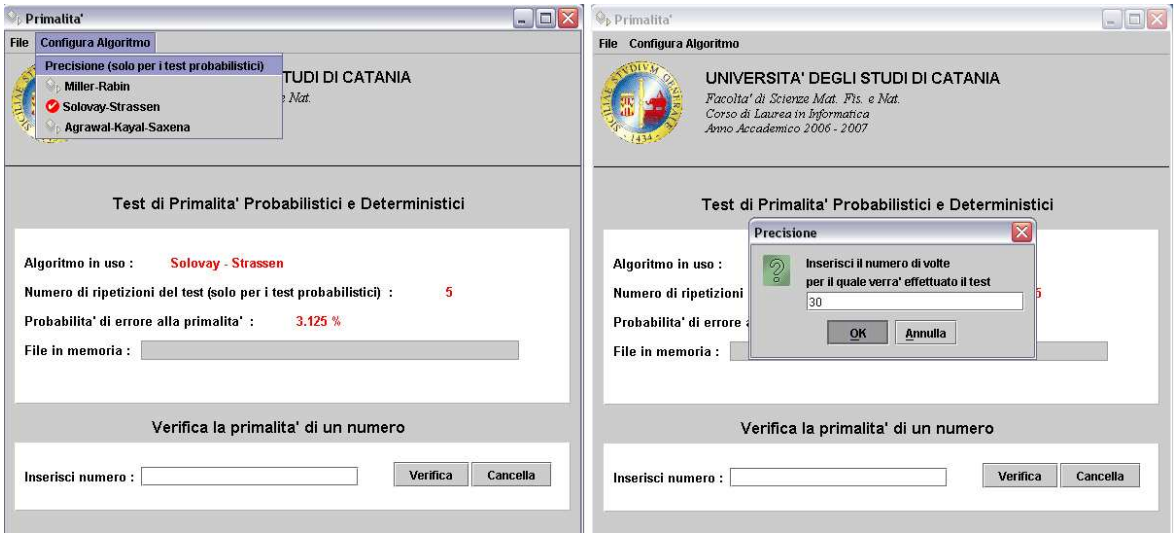
1. il metodo *jacobi(x, y, t)* della classe *SolovayStrassen*, che calcola il simbolo di jacobi $\left(\frac{x}{y}\right)$ (vedi Teoria)
2. i campi nu_0 e m_0 della classe *MillerRabin*, destinate a contenere quei valori tali che $n = 2^{nu\_0} \cdot m\_0$ con m_0 dispari.

La classe *aks*, esegue l'algoritmo AKS sul campo n, con il metodo *eseguialgoritmoaks()* e inserisce il risultato e il tempo di elaborazione rispettivamente su *output* e *tempo*. Come sappiamo dalla teoria, l'algoritmo deve stabilire al primo passo se il campo n è una potenza perfetta o meno. Questo lavoro è svolto dalla classe *Ispower*. In pratica, la classe *power* determina con *ricercabase()* il $\lfloor n^{1/b} \rfloor$ per ogni n e b passati come parametri al metodo *assegnavalore(..)*, mentre la classe *Ispower*, attraverso *trovapotenza()*, fa variare il valore di b alla ricerca (se esiste) di quel valore tale che $\lfloor n^{1/b} \rfloor^b = n$. Inoltre, l'algoritmo può richiedere il calcolo della funzione totiente e questa operazione è svolta dal metodo *calcolaphi()* della classe *phifunction*. Questo metodo, a sua volta, ha la necessità di stabilire la primalità di certi numeri (vedi definizione di funzione totiente) e quindi utilizza un algoritmo banale ed esponenziale di primalità implementato dalla classe *primalitabruteforce*.


primalitabruteforce
-n : BigInteger
-output : int
+primalitabruteforce(numero)
+assegnavalore(numero) : void
+operazione() : void
+value() : int
phifunction
-n : BigInteger
-phi : BigInteger
+phifunction(numero)
+calcolaphi() : void
+value() : BigInteger
1..*
1
0..1
1
aks
-n : BigInteger
-output : int
+tempo : float
+aks()
+aks(numero)
+assegnavalore(numero) : void
+eseguialgoritmoaks() : void
+value() : int
1
1
power
-n : BigInteger
-b : BigInteger
-q : BigInteger
-x_0 : BigInteger
-m : int
+power()
+assegnavalore(param_n, param_b) : void
+ricercabase() : void
+value() : BigInteger
1..*
1
Ispower
+n : BigInteger
+output : int
+Ispower(param_n)
+trovapotenza() : void
+value() : int

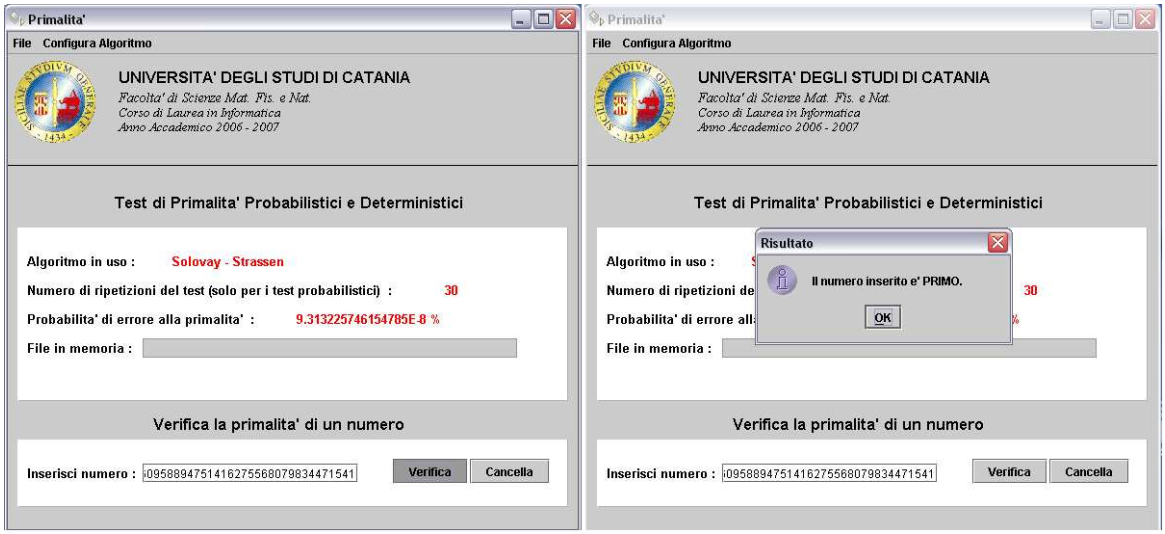
A.4 Validazione

Procediamo alla verifica del software mostrando, con esempi pratici, come funziona il programma e se produce i risultati previsti. La prima operazione che si deve fare è scegliere l'algoritmo da usare attraverso il menù "Configura Algoritmo ". Sempre dallo stesso menù è possibile impostare la precisione degli algoritmi probabilistici : cliccando su "Precisione (solo per i test probabilistici)"si aprirà un JOptionButton e si potrà inserire il numero di volte per il quale verrà ripetuto un test probabilistico. Ecco come si imposta la precisione a 30.



A questo punto, possiamo introdurre un numero usando il JTextFied presente nella parte bassa del JFrame e verificarne la primalità cliccando sul bottone "Verifica". Supponiamo di voler testare il numero :

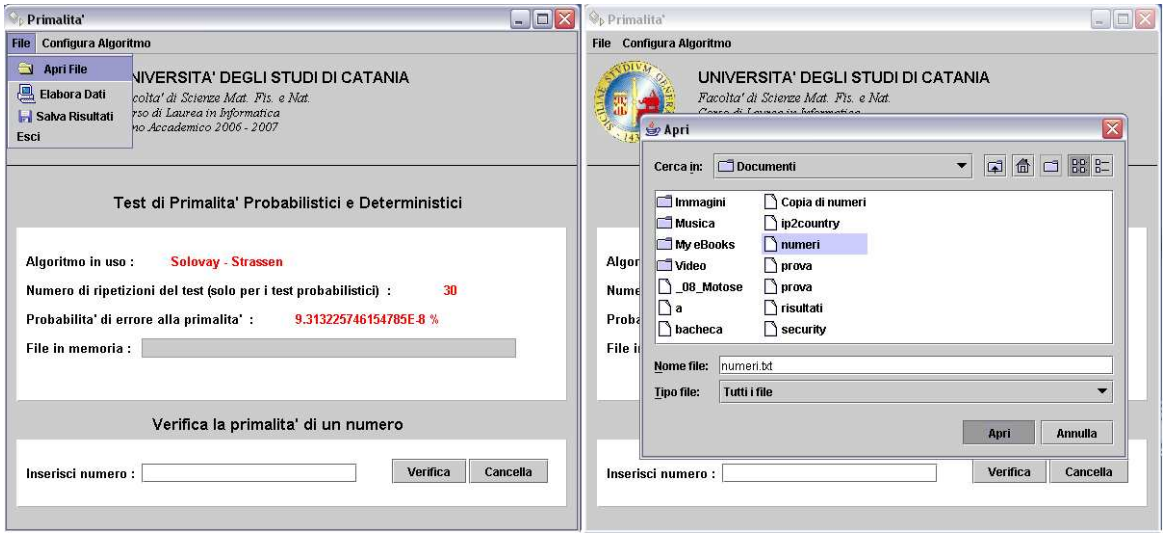
3416125484651880824649192099235033527914617803403701054777894769025318772710175775793367341953921714553521186823810932269184143033013016793222402163346724187114477766138018341566351560693965548954433712778344922065660960958894751416275568079834471541



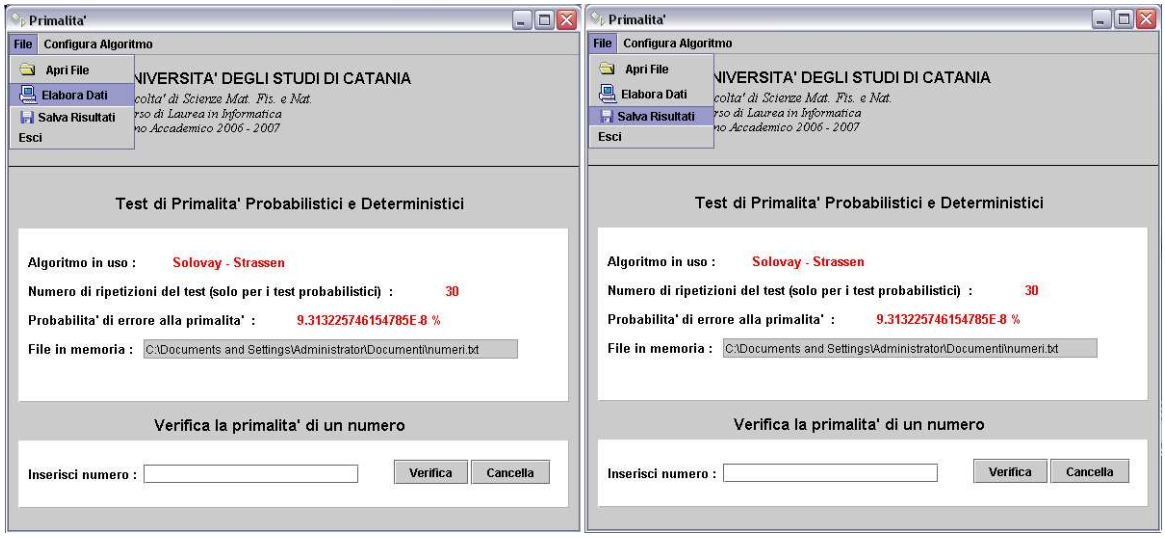
Dopo aver seguito le indicazioni, il programma ci segnalerà "Il numero inserito è PRIMO". Questo significa, controllando le impostazioni scelte precedentemente, che per l'algoritmo Solovay-Strassen il numero inserito è primo con probabilità di errore pari a $9,3 \cdot 10^{-8}\%$. Supponiamo, adesso, di voler verificare la primalità di una serie di numeri. Attraverso un text-editor scriviamo una sequenza di numeri, separati da un carattere di ritorno a capo, in un file. Ad esempio, costruiamo un file di nome "numeri.txt"con questi numeri (Per ragioni tipografiche, uso uno spazio invece del ritorno a capo) :

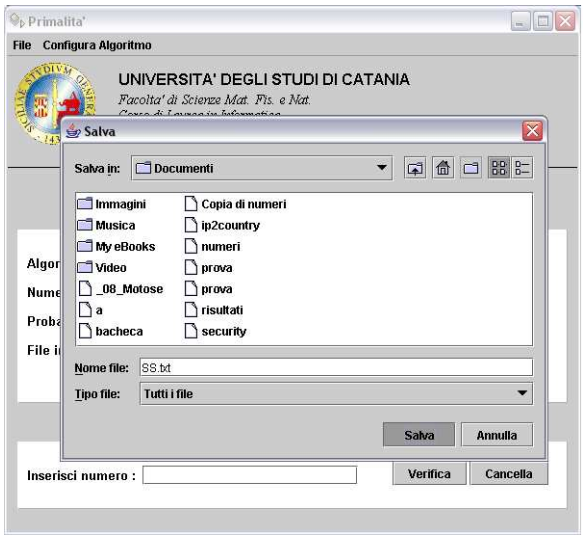
2 3 4 5 6 13 17 21 24 33 37 41 47 53 307 402 8546 6773 8374 10345 11701 12348

A questo punto, dal menù "File", clicchiamo su "Apri File "e selezioniamo il file "numeri.txt".



I dati sono stati così caricati in memoria e sono pronti per essere elaborati con l'opzione "Elabora Dati". Successivamente, cliccando su "Salva Risultati"è possibile indicare il nome del file su cui salvare i risultati (nel nostro caso "SS.txt") e procedere effettivamente all'operazione cliccando su "Salva risultati".





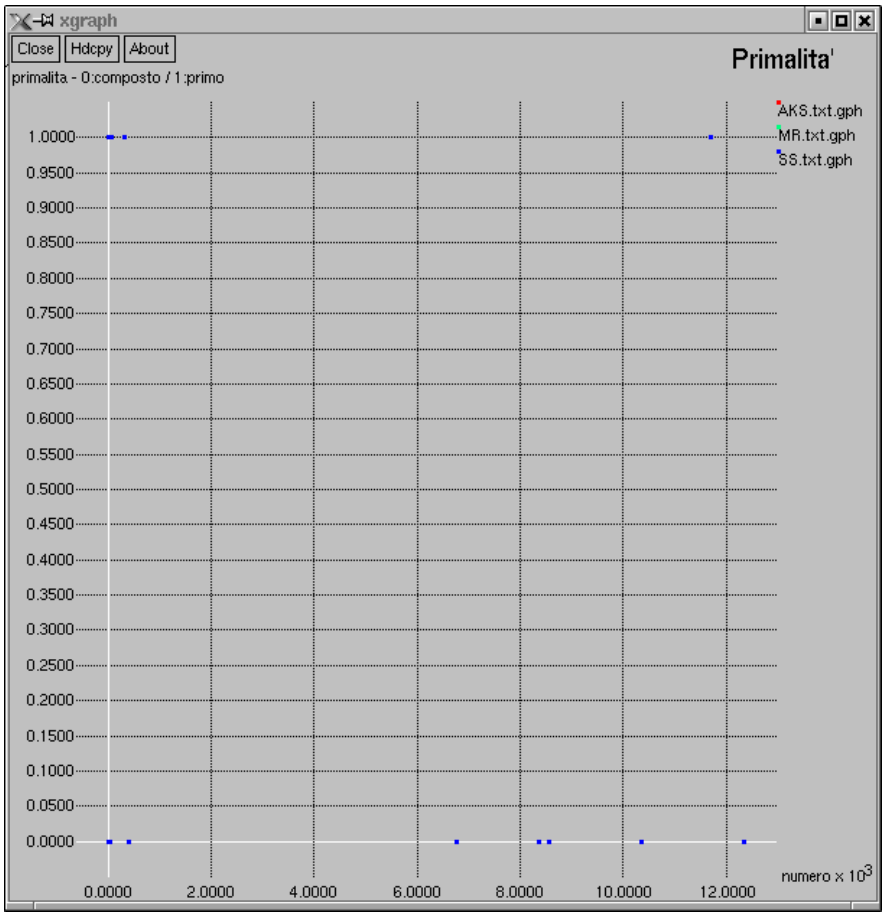
Il file di risultati conterrà righe della forma : numero, carattere TAB, 1 o 0 (primo/composto), carattere TAB, tempo di elaborazione in secondi, carattere RETURN. Ecco come si presenterà il file "SS.txt":

2	1	0.0050
3	1	0.0080
4	0	0.0
5	1	0.0030
6	0	0.0
13	1	0.0050
17	1	0.0080
21	0	0.0
24	0	0.0
33	0	0.0
37	1	0.0050
41	1	0.0050
47	1	0.0040
53	1	0.0020
307	1	0.0090
402	0	0.0
8546	0	0.0
6773	0	0.0
8374	0	0.0
10345	0	0.0
11701	1	0.01
12348	0	0.0

Attraverso uno script bash (che richiama al suo interno il tool grafico xgraph) è possibile fare un confronto dei tre algoritmi. Sulla stessa lista di numeri presente nel file "numeri.txt", applichiamo gli algoritmi e salviamo i risultati su tre file diversi. Immaginiamo che questi file siano : MR.txt, SS.txt, AKS.txt. Attraverso il comando

$$./graficarisultati.sh\ 1\ MR.txt\ SS.txt\ AKS.txt$$

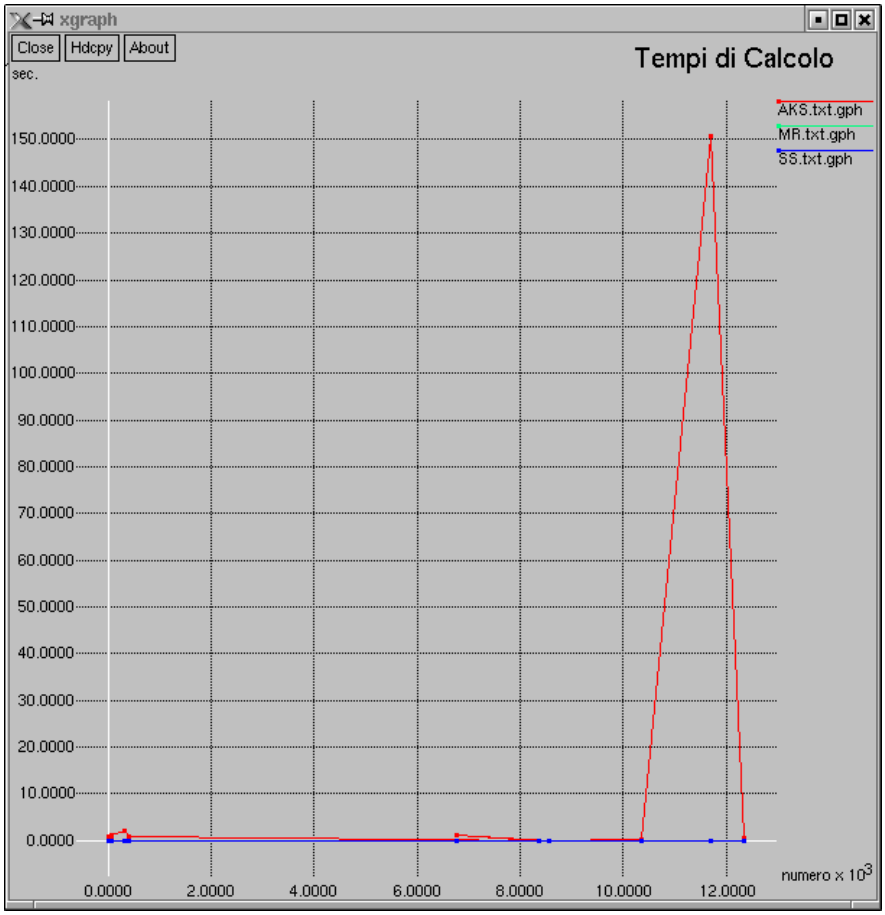
osserviamo che gli algoritmi producono gli stessi risultati, e ciò dimostrato dal fatto che i punti di MR.txt e AKS.txt sono sovrascritti dai punti SS.txt.



Mentre con il comando

$$./graficarisultati.sh\ 2\ MR.txt\ SS.txt\ AKS.txt$$

grafichiamo i tempi di elaborazione dei tre algoritmi. Osserviamo che gli algoritmi probabilistici sono più veloci dell'algoritmo AKS. Quest'ultimo, addirittura, su un numero come 11701 può impiegare ben 150 secondi (Pentium Mobile 2.2GHz, 512Mb Ram).



Provando ripetutamente gli algoritmi si può concludere che

1. Gli algoritmi producono i risultati "previsti".

2. I tempi di elaborazione degli algoritmi probabilistici sono eccellenti (anche con precisioni elevate)

3. L'importante risultato teorico raggiunto con l'algoritmo AKS, non può ritenersi tale in ambito applicativo. Le operazioni polinomiali rallentano notevolmente l'algoritmo tanto da impedirne la sua applicabilità a grandi numeri.

Bibliografia

[1] Victor Shoup, *A computional Introduction to Number Theory and Algebra.*
Cambridge University Press (1994).
Versione elettronica disponibile all'indirizzo :
http://www.shoup.net/ntb/ntb-v1.pdf.

[2] Hardy & Wright, *Introduction to the theory of Numbers.*
Versione elettronica disponibile all'indirizzo :
http://modular.fas.harvard.edu/scans/paper/hardy/Hardy-Wright-Theory_of_Numbers.pdf.

[3] H. Cormen, E. Leiserson, L. Rivest, *Introduzione agli Algoritmi.*
Jackson Libri (2003).

[4] Louis Monier, *Evaluation and Comparison of two efficient probabilistic primality testing algorithms.*
Theorical Computer Science 12 (1980) pag. 97-108.

[5] Burt Rosenberg, *The Solovay-Strassen Primality Test.*
Paper disponibile all'indirizzo :
http://www.cs.miami.edu/ burt/learning/Css609.022/jacobi.pdf.

[6] Roberto Volpe, *Note sulla Teoria dei Residui Quadratici.*
Paper disponibile all'indirizzo :
http://arxiv.org/ftp/math/papers/0504/0504335.pdf.

[7] Shabnam Akhtari, *On the Cyclotomic Polynomial with +1 or -1 Coefficients.*
B.Sc., Sharif university of Techonology (2002)
Versione elettronica disponibile all'indirizzo :
http://www.math.sfu.ca/numthry/report/2004jul31.pdf.

[8] Kaoru Motose, *On values of cyclotomic polynomials. V*
Paper disponibile all'indirizzo :
http://www.math.okayama-u.ac.jp/mjou/mjou45/_08_Motose.pdf.

[9] M. Nair, *On Chebyshev-Type Inequalities for Primes.*
The American Mathematical Monthly, Vol.89, No. 2(1982)
Una versione giapponese dei teoremi utilizzati è disponibile all'indirizzo :
http://idm.s9.xrea.com/etc/lcm_lbound.pdf.

[10] Michiel Smid, *Primality testing in polynomial time.*
Paper disponibile all'indirizzo :
http://citeseer.ist.psu.edu/smid03primality.html.

[11] Michael Stay, *Primes is in P, Slowly.*
Paper disponibile all'indirizzo :
http://math.ucr.edu/ mike/primes.ps.

[12] M. Agrawal, N. Kayal & N. Saxena, *PRIMES is in P.*
Annals of Mathematics, 160 (2004), pag. 781-793
Versione elettronica disponibile all'indirizzo :
http://www.math.princeton.edu/ annals/issues/2004/Sept2004/Agrawal.pdf.