Teoria dei numeri computazionale

I gruppi di base

Poniamo $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$ che denota l'insieme di interi. Poniamo $\mathbb{Z}_+ = \{1, 2, \dots\}$ che denota l'insieme degli interi positivi e $\mathbb{N} = \{0, 1, 2, \dots\}$ l'insieme degli interi non negativi.

Interi mod N

Se a, b sono interi, non entrambi zero, allora il loro massimo comun divisore, indicato con $MCD(a, b)$, è il più grande intero d tale che d divide a e d divide b. Se $MCD(a, b) = 1$ allora diciamo che a e b sono primi tra loro. Se considero a ed N sono interi con $N > 0$ allora ci sono interi unici r, q tali che $a = Nq + r$ e $0 \leq r < N$. Chiamiamo r il resto della divisione di a per N, e lo indichiamo con $\mod N$. Notiamo che l'operazione $a \mod N$ è definita sia per valori negativi che non negativi di a, ma solo per valori positivi di N. (Quando a è negativo, anche il quoziente q sarà negativo, ma il resto r deve essere sempre compreso nell'intervallo indicato $0 \leq r < N$). Se a, b sono interi qualsiasi e N è un intero positivo, scriviamo $a \equiv b\ (mod\ N)$ se $a\ mod\ N = b\ mod\ N$. Associamo a qualsiasi intero positivo N i seguenti due insiemi:

$$\mathbb{Z}_N = \{0, 1, \dots, N-1\}$$
$$\mathbb{Z}_N^* = \{i \in \mathbb{Z} : 1 \leq i \leq N-1 \land MCD(i, N) = 1\}$$

Il primo insieme è detto **insieme di interi mod N**. La sua dimensione è N, e contiene esattamente gli interi che sono possibili valori di un *mod N* come intervalli su $\mathbb{Z}$. Definiamo la funzione di **Eulero Phi** $\phi : \mathbb{Z}_+ \to \mathbb{N}$ per $\phi(N) = |\mathbb{Z}_N^*|$ per ogni $N \in \mathbb{Z}_+$. Cioè, $\phi(N)$ è la dimensione dell'insieme $\mathbb{Z}_N^*$.

Gruppi

Sia G un insieme non vuoto, e sia $\cdot$ un'operazione binaria su G. Ciò significa che per ogni due punti $a, b \in G$, viene definito un valore $a \cdot b$.

Definizione 1.1 Sia G un insieme non vuoto e sia $\cdot$ indichiamo un'operazione binaria su G. Diciamo che G è un gruppo se ha le seguenti proprietà:

1. **Chiusura**: Per ogni $a, b \in G$ è il caso che anche $a \cdot b$ sia in G.
2. **Associatività**: Per ogni $a, b, c \in G$ è il caso che $(a \cdot b) \cdot c = a \cdot (b \cdot c)$.
3. **Identità**: esiste un elemento $1 \in G$ tale che $a \cdot 1 = 1 \cdot a = a$ per ogni $a \in G$.
4. **Invertibilità**: Per ogni $a \in G$ esiste un unico $b \in G$ tale che $a \cdot b = b \cdot a = 1$.

L'elemento b nella condizione di invertibilità è indicato come l'inverso dell'elemento a, ed è denotato a^{-1} .

Torniamo ora agli insiemi che abbiamo definito sopra e facciamo un'osservazione sulla loro struttura di gruppo. Sia N un intero positivo. L'operazione di addizione modulo N prende in ingresso due qualsiasi interi a, b e restituisce $(a+b) \mod N$. L'operazione di moltiplicazione modulo N prende in input due interi qualsiasi a, b e restituisce $a \cdot b \mod N$.

Proposizione 1.2 (Algoritmo di divisione) Se $a, b \in \mathbb{Z}$, $b \neq 0$, allora esistono $q, r \in Z$ (quoziente e resto) tali che $a = bq + r$ con $0 \leq r \leq |b|$.

DIMOSTRAZIONE. Supponiamo dapprima che $a \geq 0$ e procediamo per induzione su a. Per $a = 0$ la proprietà è vera, basta prendere $q = r = 0$. Supposta vera la proprietà per a, cioè supponiamo che $a = bq + r$ con $0 \leq r \leq |b|$ e proviamola per $a + 1$. Ma da $a = bq + r$ abbiamo $a + 1 = bq + r + 1$; per cui se $r + 1 < |b$ la proprietà è vera per $a + 1$; se invece $r + 1 = |b|$, cioè $r + 1 = \pm b$ allora avremo $a + 1 = b(q \pm 1) + 0$ ed anche in tal caso la proprietà è vera per $a + 1$. Così per l'ipotesi induttiva l'algoritmo di divisione è vero per ogni a naturale. Infine, se $a < 0$ la proprietà sarà vera per $-a$ e quindi $-a = bq + r$ con $0 \leq r < |b|$. Allora $a = b(-q) - r$, per cui se $r = 0$ abbiamo la tesi, se invece $r > 0$, scriveremo $a = b(-q) - |b| + |b| - r$, cioè $a = b(-q \pm 1) + r'$ con $r' = |b| - r < |b|$.

Quoziente e resto sono unici.

Proposizione 1.3 (Algoritmo Euclideo) Siano $a, b \in \mathbb{Z}$, $b \neq 0$, consideriamo le seguenti divisioni successive.

$$\begin{gathered}
a = bq_1 + r_1, \text{ con } 0 \leq r_1 < |b|, \text{ e se } r_1 \neq 0, \\
b = r_1 q_2 + r_2, \text{ con } 0 \leq r_2 < r_1, \text{ e se } r_2 \neq 0, \\
r_1 = r_2 q_3 + r_3, \text{ con } 0 \leq r_3 < r_2, \text{ e se } r_3 \neq 0, \\
\dots \dots \dots \\
r_{n-2} = r_{n-1} q_n + r_n, \text{ con } 0 \leq r_n < r_{n-1}, \text{ e se } r_n \neq 0, \\
r_{n-1} = r_n q_{n+1};
\end{gathered}$$

Allora r_n (ultimo resto non nullo) è il cercato massimo comune divisore.

DIMOSTRAZIONE. Euclide osservò che tale procedura ha termine in quanto $r_1 < |b|$, $r_2 < r_1$, ecc. per cui dopo al più b divisioni successive il resto deve essere 0. Per provare che tale r_n è il $MCD(a, b)$ cominciamo ad osservare che dall'ultima eguaglianza segue $r_n | r_{n-1}$; dalla penultima segue che $r_n | r_{n-2}$; così risalendo dalla terza deduco $r_n | r_1$, dalla seconda $r_n || b|$ ed infine dalla prima $r_n | a$. In definitiva, r_n soddisfa la prima condizione richiesta dal $MCD(a, b)$.
D'altra parte, se d è un divisore comune ad a e b dalla prima divisione deduciamo che $d | r_1$, quindi dalla seconda deduciamo $d | r_2$ e dalla terza $d | r_3$, così continuando quando arriveremo all'ultima divisione troveremo che $d | r_n$ e la seconda condizione per il $MCD(a, b)$ resta verificata. Per cui $r_n = MCD(a, b)$.

Proposizione 1.4 (Identità di Bézout) Se $d = MCD(a, b)$ allora si possono trovare λ e μ in $\mathbb{Z}$ tali che $d = \lambda a + \mu b$.

DIMOSTRAZIONE. Basta utilizzare le divisioni successive dell'algoritmo euclideo partendo dall'ultima e risalendo sino alla prima.

Esempio

$MCD(750, 72)$

$$
\begin{aligned}
750 &= -72 \cdot (-10) + 30, \\
-72 &= 30 \cdot (-3) + 18, \\
30 &= 18 \cdot 1 + 12, \\
18 &= 12 \cdot 1 + 6 \\
12 &= 6 \cdot 2
\end{aligned}
$$

L'identità di Bézout è la seguente:

$$
\begin{gathered}
6 = 18 - 1 \cdot 12, \\
6 = 18 - [30 - 1 \cdot 18] = 2 \cdot 18 - 30, \\
6 = 2 \cdot [-72 + 3 \cdot 30] - 30 = 2 \cdot (-72) + 5 \cdot 30, \\
6 = 2 \cdot (-72) + 5 \cdot [750 + 10 \cdot (-72)] = 5 \cdot 750 + 52 \cdot (-72), \\
6 = 5 \cdot 750 + 52 \cdot (-72).
\end{gathered}
$$

Per cui si ha $\lambda = 5$ e $\mu = 52$.

Osservazione 1.5 Sia N un intero positivo. Allora $\mathbb{Z}_N$ è un gruppo sotto addizione modulo N e $\mathbb{Z}_N^*$ è un gruppo sotto moltiplicazione modulo N.

In $\mathbb{Z}_N$, l'elemento identità è 0 e l'inverso di a è $-a \mod N = N - a$. In $\mathbb{Z}_N^*$, l'elemento identità è 1 e l'inverso di a è $a \cdot b \in \mathbb{Z}_N^*$ tale che $ab \equiv 1$ $(mod\ N)$. Questa cosa è vera perché ci si restringe alle classi i cui rappresentanti sono coprimi con N, per dimostrare tale proprietà è necessaria l'**identità di Bézout**: se a è coprimo con N, esistono due interi x, y tali che:

$$
\begin{gathered}
ax + Ny = 1 \\
ax \equiv 1 \mod N
\end{gathered}
$$

In qualsiasi gruppo, possiamo definire un'operazione di elevazione a potenza che associa a qualsiasi $a \in G$ e a qualsiasi intero i un elemento di gruppo che denotiamo a^i, definito come segue. Se $i = 0$ allora a^i è definito come 1, l'elemento di identità del gruppo. Se $i > 0$ allora:

$$a^i = a \cdot a \cdot \ldots \cdot a \text{ ripetuto i-volte}$$

Se i è negativo, allora possiamo definire $a^i = (a^{-1})^j$, con $j = |-i|$, per cui si ha:

$$a^i = a^{-1} \cdot a^{-1} \cdot \ldots \cdot a^{-1} \text{ ripetuto j-volte}$$

Con queste definizioni in atto, possiamo manipolare gli esponenti nel modo in cui siamo abituati con i numeri ordinari.

È consuetudine nella teoria dei gruppi chiamare la dimensione di un gruppo G il suo ordine. Cioè, l'ordine di un gruppo G è $|G|$, il numero di elementi in esso. Utilizzeremo spesso il seguente fatto di base.
Se un qualsiasi elemento di gruppo viene elevato all'ordine del gruppo, il risultato è l'elemento identità del gruppo.

Osservazione 1.6 Sia G un gruppo e sia $m = |G|$ il suo ordine. Allora $a^m = 1\ \forall a \in G$.

Ciò significa che il calcolo negli indici di gruppo può essere eseguito modulo m:

Proposizione 1.7 Sia G un gruppo e sia $m = |G|$ il suo ordine. Allora $a^i = a^{i \mod m}\ \forall a \in G \wedge \forall i \in \mathbb{Z}$.

DIMOSTRAZIONE. La dimostrazione di tale preposizione segue dall'Osservazione 1.6.

Poniamo $0 \leq i < m$ in questo caso si ha che $a^i = a^{i \mod m}$ è sempre vera.
Poniamo $i = m$ in questo caso si ha che $a^m = a^{m \mod m} = 1$ e anche in questo caso la proprietà resta vera.
Poniamo $i \geq m$ in questo caso procediamo per induzione, per $i = m$ abbiamo già visto che la proprietà è vera, quindi consideriamo $i = m + 1$ per cui si ha $a^{m+1} = a^{m+1 \mod m}$ da cui segue $a^m a = a^{m \mod m} a \to a = a$, per cui continua a valere la proprietà.
Poniamo $i \leq 0$ in questo caso procediamo per induzione, per $i = 0$ abbiamo già visto che la proprietà è vera, quindi consideriamo $i = -1$ per cui si ha $a^{-1} = a^{-1 \mod m}$ da cui segue $a^{-1} = a^{m-1} \to a^{-1} = a^{-1}$, per continua a valere la proprietà.

Se G è un gruppo, un insieme $S \subseteq G$ è detto sottogruppo se è un gruppo a sé stante, sotto la stessa operazione di quella per cui G è un gruppo. Se sappiamo già che G è un gruppo, c'è un modo semplice per verificare se S è un sottogruppo: se e solo se $x \cdot y^{-1} \in S$ per ogni $x, y \in S$. Qui y^{-1} è l'inverso di y in G.

Osservazione 1.8 Sia G un gruppo e sia S un sottogruppo di G. Allora l'ordine di S divide l'ordine di G.

Algoritmi

Normalmente ignoriamo il costo delle operazioni di base (ad es. addizioni e moltiplicazioni). Con numeri "crittografici" tali costi non possono essere ignorati.

- $a \mod N$ costa $O(|a||N|)$;
- sommare due interi di k bit richiede $O(k)$ operazioni binarie;
- moltiplicare due interi di k bit richiede $O(k^2)$ operazioni binarie;
- calcolare $a^m \mod N$, $|N| = |a| = k$, costa $O(|m|k^2)$.

Algoritmi di divisione intera e modulo

Definiamo la funzione di divisione tra interi prendendo in input due interi a, N, con $N > 0$, e restituendo il quoziente e il resto ottenuti dividendo a per N. Cioè, la funzione restituisce (q, r) tale che $a = qN + r$ con $0 \leq r < N$. Indichiamo con `INT-DIV` un algoritmo che implementa questa funzione.
L'algoritmo utilizza il metodo di divisione standard che abbiamo imparato a scuola, che risulta essere eseguito in un tempo proporzionale al prodotto delle lunghezze binarie di a e N.
Vogliamo anche un algoritmo che implementi la funzione mod, prendendo input interi a, N con $N > 0$ e restituendo un $\mod N$. Questo algoritmo, denominato `MOD`, può essere implementato semplicemente chiamando `INT-DIV(a, N)` per ottenere (q, r), e quindi restituendo solo il resto r.

Algoritmo di Euclide esteso

In aritmetica e nella programmazione l'algoritmo esteso di Euclide è un'estensione dell'algoritmo di Euclide che calcola non solo il massimo comun divisore tra due numeri a, b, ma anche i coefficienti dell'identità di Bézout.

Supponiamo che a, b siano interi, entrambi non nulli. Un fatto fondamentale sul massimo comun divisore di a e b è che è il più piccolo elemento positivo dell'insieme di tutte le combinazioni lineari intere di a e b.

$$\{\lambda a + \mu b : \lambda, \mu \in \mathbb{Z}\}$$

In particolare, se $d = MCD(a, b)$ allora esistono interi λ, μ tali che $d = \lambda a + \mu b$. (Nota che λ o μ potrebbero essere negativi).

Oltre al MCD stesso, troveremo utile poter calcolare questi pesi λ, μ. Questo è ciò che fa l'algoritmo di Euclide esteso `EXT-GCD`: dati a, b come input, restituisce (d, λ, μ) tale che $d = MCD(a, b) = \lambda a + \mu b$. L'algoritmo stesso è un'estensione del classico algoritmo di Euclide per il calcolo del MCD e la descrizione più semplice è ricorsiva. Ora lo forniamo e poi ne discutiamo la correttezza e il tempo di esecuzione. L'algoritmo accetta in input qualsiasi numero intero a, b, entrambi non nulli.

```
EXT-GCD(a, b):
    if (b == 0) then return (a, 1, 0)
    else
        (q, r) <- INT-DIV(a, b)
        (d, x, y) <- EXT-GCD(b, r)
        lambda <- y
        mu <- x - qy
        return (d, lambda, mu)
```

Il caso base è quando $b = 0$. Se $b = 0$ allora sappiamo per assunzione che $a \neq 0$, quindi $MCD(a, b) = a$, e poiché $a = a(1) + b(0)$, i pesi sono 1 e 0. Se $b \neq 0$ allora possiamo dividere per esso, e dividiamo a per esso per ottenere un quoziente q e il resto r. Per la ricorsione usiamo il fatto che $MCD(a, b) = MCD(b, r)$. La chiamata ricorsiva produce quindi $d = MCD(a, b)$ insieme ai pesi x, y tali che $d = bx + ry$. Notando che $a = bq + r$ abbiamo la conferma che i valori assegnati ad λ, μ sono corretti.

$$d = bx + ry = bx + (a - bq)y = ay + b(x - qy) = a\lambda + n\mu$$

Il tempo di esecuzione di questo algoritmo è $O(|a| \cdot |b|)$.

Algoritmo per l'inverso modulare

Serve per il calcolo dell'inverso moltiplicativo di a nel gruppo $\mathbb{Z}_N^*$. Ovvero, su input $N > 0$ e $a \in \mathbb{Z}_N^*$, l'algoritmo `MOD-INV` restituisce b tale che $a \cdot b \equiv 1 \ (mod\ N)$. Il metodo è abbastanza semplice:

```
MOD-INV(a, N):
    (d, x, y) <- EXT-GCD(a, N)
    b <- x mod N
    return b
```

Poiché $a \in \mathbb{Z}_N^*$ sappiamo che $MCD(a, N) = 1$. L'algoritmo `EXT-GCD` garantisce quindi che $d = 1$ e $1 = ax + Ny$. Poiché $N \mod N = 0$, abbiamo $ax \equiv 1 \mod N$, e quindi $b = x \mod N$ è il valore giusto da restituire. Il costo di tale procedura è $O(|a| \cdot |N|)$.

Teorema Cinese del resto (CRT)

È un metodo per risolvere un certo tipo di congruenze.

Siano $m_1, \ldots, m_n$ coprimi e $a_1, \ldots, a_n$ tali che:

$$x = a_1 \mod m_1$$
$$\ldots$$
$$x = a_n \mod m_n$$

Il teorema cinese del resto ci assicura che tale sistema ha un'unica soluzione modulo $M = m_1, \ldots, m_n$ e ci dice come calcolarla.

Tale soluzione è:

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

dove $M_i = \frac{M}{m_i}$ e $y_i = M_i^{-1} \mod m_i$

Gruppi ciclici e generatori

Sia G un gruppo, sia 1 il suo elemento di identità, e sia $m = |G|$ l'ordine di G. Se $g \in G$ è un qualsiasi membro del gruppo, l'ordine di g è definito come l'intero meno positivo n tale che $g^n = 1$. Poniamo

$$\langle g \rangle = \{g^i : i \in \mathbb{Z}_n\} = \{g^0, g^1, \ldots, g^{n-1}\}$$

con cui indichiamo l'insieme degli elementi di gruppo generati da g. Un fatto che non dimostriamo, ma è facile da verificare, è che questo insieme è un sottogruppo di G. L'ordine di questo sottogruppo (che, per definizione, è la sua dimensione) è proprio l'ordine di g. L'Osservazione 1.8 ci dice che l'ordine n di g divide l'ordine m del gruppo.

Definizione 1.9 Un elemento g del gruppo si dice generatore di G se $g = G$, o, equivalentemente, se il suo ordine è m.

Definizione 1.10 Se G ammette un generatore esso è detto ciclico.

Se g è un generatore di G allora per ogni $a \in G$ esiste un unico intero $i \in \mathbb{Z}_m$ tale che $g^i = a$. Questo i è chiamato **logaritmo discreto** di a in base g, e lo denotiamo con $DLog_{G,g}(a)$. Quindi, $DLog_{G,g}(\cdot)$ è una funzione che mappa G in $\mathbb{Z}_m$, e inoltre questa funzione è una **biiezione**, cioè uno a uno. La funzione di $\mathbb{Z}_m$ a G definita da $i \to g^i$ è chiamata **funzione di elevamento a potenza discreta**, e la funzione logaritmo discreto è l'inversa della funzione di elevamento a potenza discreta.

Esempio:

Sia $p = 11$, che è primo. Allora $\mathbb{Z}_{11}^* = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ che ha ordine $p - 1 = 10$. Consideriamo i sottogruppi generati dagli elementi 2 e 5. A questo punto li eleviamo a potenze di $i = 0, \ldots, 9$ ottenendo:

i	0	1	2	3	4	5	6	7	8	9
$2^i \mod 11$	1	2	4	8	5	10	9	7	3	6
$5^i \mod 11$	1	5	3	4	9	1	5	3	4	9

Guardando quali elementi compaiono nella riga corrispondente a 2 e 5, rispettivamente, possiamo determinare i sottogruppi che questi elementi di gruppo generano:

$$\langle 2 \rangle = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$$
$$\langle 5 \rangle = \{1, 3, 4, 5, 9\}$$

Poiché 2 è uguale a $\mathbb{Z}_{11}^*$, l'elemento 2 è un generatore. Poiché esiste un generatore, $\mathbb{Z}_{11}^*$ è ciclico. D'altra parte, $5 \neq Z_{11}^*$, quindi 5 non è un generatore. L'ordine di 2 è 10, mentre l'ordine di 5 è 5. Nota che questi ordini dividono l'ordine 10 del gruppo. La tabella ci permette anche di determinare i logaritmi discreti in base 2 dei diversi elementi del gruppo:

a	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
$DLog_{\mathbb{Z}_{11}^*,2}(a)$	0	1	8	2	4	9	7	3	6	5

La funzione di elevamento a potenza discreta è congetturata essere unidirezionale (il che significa che la funzione logaritmica discreta è difficile da calcolare) per alcuni gruppi ciclici G. Per questo motivo spesso cerchiamo gruppi ciclici per l'uso crittografico. Definiamo formalmente ciò che è stato visto fino ad ora.

Se $(G, \cdot)$ è un gruppo ed $X \subseteq G$ un suo sottoinsieme, in generale non un sottogruppo, si può considerare in un certo senso il "più piccolo" sottogruppo che contiene X. Precisamente,

Definizione 1.11 Siano $(G, \cdot)$ un gruppo ed $X \subseteq G$ un suo sottoinsieme, si dice sottogruppo generato da X, e sarà indicato con $G(X)$ l'intersezione di tutti i sottogruppi contenenti X, cioè:

$$G(X) = \bigcap_{i \in I} S_i$$

dove $\{S_i\}_I$ denota la famiglia dei sottogruppi di G che contengono X.

Se $G(X) = G$ si dirà che X è un sistema di generatori per G. Il caso più semplice è più interessante si ha quando $X = \{a\}$ è costituito da un solo elemento. In tal caso assumeremo la seguente definizione.

Definizione 1.12 Un gruppo G si dice **ciclico** se esiste un elemento $a \in G$ tale che $G = G(a)$.

Evidentemente quando G è ciclico i suoi elementi sono facilmente esprimibili. Infatti, se $G = G(a)$ allora:

$$G = \{a^i | \forall i \in \mathbb{Z}\}$$

Questo fatto ci dice, tra l'altro, che il più piccolo sottogruppo contenente un elemento a è l'insieme di tutte le sue potenze, basta osservare che $a^i \cdot a^j = a^{i+j}$ e che se un sottogruppo contiene a deve contenere a^{-1} e quindi tutte le sue potenze con esponenti in $\mathbb{Z}$.

Un gruppo ciclico con un numero finito di elementi si dice **ciclico finito** altrimenti si dirà **ciclico infinito**.

Proposizione 1.13 Sia G un gruppo ciclico finito generato da $a \neq e$, con e l'elemento identità del gruppo. Allora esiste $m \in \mathbb{N}^*$ tale che $a^m = e$.

DIMOSTRAZIONE. Visto che $G = G(a)$ tutte le potenze di a stanno in G, ma essendo questo finito non possono essere tutte distinte queste potenze, sicché esisteranno $i, j \in \mathbb{Z}, i \neq j$, tali che $a^i = a^j$. Supponiamo $i > j$; allora $m = i - j > 0$. Ma da $a^i = a^j$, moltiplicando ambo i membri per a^{-j} si ottiene $a^{i-j} = a^{j-j}$ cioè $a^m = e$.

Alla luce di questo risultato si ha:

Proposizione 1.14 Sia $G = G(a)$ un gruppo ciclico finito e poniamo $n = \min\{m \in \mathbb{N}^* | a^m = e\}$. Allora $G = \{e = a^0, a, a^2, \ldots, a^{n-1}\}$, quindi G ha ordine n.

DIMOSTRAZIONE. Che $\{e = a^0, a, a^2, \ldots, a^{n-1}\} \subseteq G$ è ovvio. Viceversa, se $g \in G$ visto che $G = G(a)$, $g = a^t$. Usando l'algoritmo di divisione tra t ed n si ha che $t = qn + r$ con $0 \leq r < n$, per cui:

$$g = a^t = a^{qn+r} = (a^n)^q a^r = ea^r = a^r$$

Infine, per concludere che l'ordine di G è proprio n, dobbiamo mostrare che tutti gli elementi in $\{e = a^0, a, a^2, \ldots, a^{n-1}\}$ sono distinti. Ed infatti, se per assurdo esistessero due interi $0 \leq i < j \leq n-1$ tali che $a^j = a^i$, moltiplicando per a^{-i} si avrebbe $a^{j-i} = e$ con $0 < j - i < n$ e questo è in contrasto con la minimalità di n.

Osservazione 1.15 Alla luce della precedente proposizione, l'ordine di un gruppo ciclico finito generato da a coincide con il più piccolo intero positivo n tale che $a^n = e$.

Osservazione 1.16 Sia p un numero primo. Allora il gruppo $\mathbb{Z}_p^*$ è ciclico.

L'operazione qui è la moltiplicazione modulo p, e la dimensione di questo gruppo è $\phi(p) = p - 1$. Questa è la scelta di gruppo più comune in crittografia.

Osservazione 1.17 Sia G un gruppo e sia $m = |G|$ il suo ordine. Se m è un numero primo, allora G è ciclico.

In altre parole, qualsiasi gruppo avente un numero primo di elementi è ciclico. Si noti che non è per questo motivo che il l'Osservazione 1.16 è vera, poiché l'ordine di Z_p^* (dove p è primo) è $p - 1$, che è pari se $p \geq 3$ e 1 se $p = 2$, e quindi non è mai un numero primo.

Ricordiamo che un campo è un insieme F dotato di due operazioni, un'addizione e una moltiplicazione. L'elemento identità dell'addizione è indicato con 0. Quando questo viene rimosso dal campo, ciò che rimane è un gruppo in moltiplicazione. Questo gruppo è sempre ciclico.

Osservazione 1.18 Sia F un campo finito, e sia $F^* = F - \{0\}$. Allora F è un gruppo ciclico sotto l'operazione di moltiplicazione di F.

Un campo finito di ordine m esiste se e solo se $m = p^n$ per qualche primo p e intero $n \geq 1$. Il campo finito di ordine p è esattamente $\mathbb{Z}_p$, quindi il caso $n = 1$ dell'Osservazione 1.18 implica l'Osservazione 1.16. Un altro caso speciale interessante dell'Osservazione 1.18 è quando l'ordine del campo è 2^n, che significa $p = 2$, ottenendo un gruppo ciclico di ordine $2^n - 1$.

Quando vogliamo usare un gruppo ciclico G in crittografia, spesso vorremmo trovargli un generatore. Il processo utilizzato consiste nel selezionare gli elementi del gruppo in un modo appropriato e quindi testare ciascun elemento scelto per vedere se si tratta di un generatore. Si devono quindi risolvere due problemi. Uno è come verificare se un dato elemento del gruppo è un generatore e l'altro è quale processo utilizzare per scegliere i generatori candidati da testare.

Sia $m = |G|$ e sia 1 l'elemento identità di G. Il modo più ovvio per verificare se un dato $g \in G$ è un generatore consiste nel calcolare i valori $g^1, g^2, g^3, \ldots$, fermandosi al primo j tale che $g^j = 1$. Se $j = m$ allora g è un generatore. Questo test tuttavia può richiedere fino a m operazioni di gruppo, il che non è efficiente, dato che i gruppi di interesse sono grandi, quindi abbiamo bisogno

di test migliori.
Il modo più ovvio per scegliere i generatori candidati è scorrere l'intero gruppo in qualche modo, testando a turno ogni elemento. Anche con un test veloce, questo può richiedere molto tempo, poiché il gruppo è numeroso. Quindi vorremmo anche modi migliori per scegliere i candidati.

Affrontiamo questi problemi a turno. Diamo prima un'occhiata nel verificare se un dato $g \in G$ è un generatore. Si vede rapidamente che calcolando tutte le potenze di g come in $g^1, g^2, g^3, \ldots$ non è necessario. Ad esempio, se abbiamo calcolato g^8 e abbiamo scoperto che questo non è 1, allora sappiamo che $g^4 \neq 1$ e $g^2 \neq 1$ e $g \neq 1$. Più in generale, se sappiamo che $g^j \neq 1$ allora sappiamo che $g^i \neq 1$ per ogni i dividendo j.
Questo ci dice che è meglio prima calcolare alte potenze di g e usarle per ridurre lo spazio degli esponenti che necessitano di ulteriori test. La seguente Proposizione individua il modo ottimale per farlo. Identifica un insieme di esponenti $m_1, \ldots, m^n$ tale che basti verificare se $g^{m_i} \neq 1$ per $i = 1, \ldots, n$. Come dimostreremo in seguito, questo set è piuttosto piccolo.

Proposizione 1.19 Sia G un gruppo ciclico e sia $m = |G|$ la dimensione di G. Sia $p_1^{\alpha_1} \cdots p_n^{\alpha_n}$ la scomposizione in fattori primi di m e sia $m_i = m/p_i$ per $i = 1, \ldots, n$. Sia $g \in G$. Allora g è un generatore di G se e solo se

$$\forall i = 1, \ldots, n : g^{m_i} \neq 1 \; (1.19)$$

dove 1 è l'elemento identità di G.

DIMOSTRAZIONE. Supponiamo prima che g sia un generatore di G. Allora sappiamo che il più piccolo intero positivo j tale che $g^j = 1$ è $j = m$. Poiché $0 < m_i < m$, deve valere che $g^{m_i} \neq 1$ per ogni $i = 1, \ldots, m - 1$.

Viceversa, supponiamo che g soddisfi la condizione dell'Equazione (1.19). Vogliamo dimostrare che g è un generatore. Sia j l'ordine di g, cioè il più piccolo intero positivo tale che $g^j = 1$. Allora sappiamo che j deve dividere l'ordine m del gruppo, cioè $m = dj$ per qualche intero $d \geq 1$. Ciò implica che $j = p_1^{\beta_1} \cdots p_n^{\beta_n}$ per alcuni interi $\beta_1, \ldots, \beta_n$ che soddisfa $0 \leq \beta_i \leq \alpha_i$ per ogni $i = 1, \ldots, n$.
Se $j < m$ allora ci deve essere qualche i tale che $\beta_i < \alpha_i$, e in tal caso j divide m_i, che a sua volta implica $g^{m_i} = 1$ (perché $g^j = 1$). Quindi l'assunzione che l'Equazione (1.19) sia vera implica che j non può essere strettamente minore di m, quindi l'unica possibilità è $j = m$, il che significa che g è un generatore.

Il numero n di termini nella scomposizione in fattori primi di m non può essere maggiore di $lg(m)$, il logaritmo binario di m. (Questo perché $p_i \geq 2$ e $\alpha_i \geq 1$ per tutti $i = 1, \ldots, n$). Quindi, per esempio, se il gruppo ha una dimensione di circa 2^{512}, sono necessari al massimo 512 test. Quindi il test è abbastanza efficiente. Si noti tuttavia che richiede la conoscenza della scomposizione in fattori primi di m.
Consideriamo ora il secondo problema che abbiamo discusso sopra, ovvero come scegliere gli elementi del gruppo candidati per il test. Sembra che ci siano poche ragioni per pensare che provare a turno tutti gli elementi del gruppo produca un generatore in un ragionevole lasso di tempo. Invece, consideriamo la scelta casuale di elementi del gruppo e poi li testiamo. La probabilità di successo in ogni prova è $|Gen(G)|/|G|$. Quindi il numero atteso di prove prima di trovare un generatore è $|G|/|Gen(G)|$. Per stimare l'efficacia di questo metodo, dobbiamo quindi conoscere il numero di generatori nel gruppo.

Proposizione 1.20 Sia G un gruppo ciclico di ordine m, e sia g un generatore di G. Allora $Gen(G) = \{g^i \in G : i \in \mathbb{Z}_m^*\}$ e $|Gen(G)| = \phi(m)$.

Cioè, avendo fissato un generatore g, un elemento di gruppo h è un generatore se e solo se il suo logaritmo discreto in base g è relativamente primo rispetto all'ordine m del gruppo. Di conseguenza, il numero di generatori è il numero di interi nell'intervallo $1, \ldots, m-1$ che sono relativamente primi a m.

DIMOSTRAZIONE. Dato che $Gen(G) = \{g^i \in G : i \in \mathbb{Z}_m^*\}$, l'affermazione sulla sua dimensione segue facilmente:

$$|Gen(G)| = |\{g^i \in G : i \in \mathbb{Z}_m^*\}| = |\mathbb{Z}_m^*| = \phi(m)$$

Dimostriamo ora che $Gen(G) = \{g^i \in G : i \in Z_m^*\}$. Innanzitutto, mostriamo che se $i \in \mathbb{Z}_m^*$ allora $g^i \in Gen(G)$. In secondo luogo, mostriamo che se $i \in \mathbb{Z}_m - \mathbb{Z}_m^*$ allora $g^i \notin Gen(G)$. Quindi prima supponiamo $i \in \mathbb{Z}_m^*$, e sia $h = g^i$. Vogliamo mostrare che h è un generatore di G. Basta mostrare che l'unico valore possibile di $j \in \mathbb{Z}_m$ tale che $h^j = 1$ è $j = 0$, quindi mostriamo ora questo. Sia $j \in \mathbb{Z}_m$ tale che $h^j = 1$. Poiché $h = g^i$ abbiamo:

$$1 = h^j = g^{ij \mod m}$$

Poiché g è un generatore, deve valere che $ij \equiv 0 \mod m$, il che significa che m divide ij. Ma $i \in \mathbb{Z}_m^*$ quindi $MCD(i, m) = 1$. Quindi deve essere che m divide j. Ma $j \in \mathbb{Z}_m$ e l'unico membro di questo insieme divisibile per m è 0, quindi $j = 0$ come desiderato.

Quindi, supponiamo $i \in \mathbb{Z}_m - \mathbb{Z}_m^*$ e sia $h = g^i$. Per mostrare che h non è un generatore basta mostrare che esiste qualche $j \in \mathbb{Z}_m$ diverso da zero tale che $h^j = 1$. Sia $d = MCD(i, m)$. La nostra ipotesi $i \in \mathbb{Z}_m - \mathbb{Z}_m^*$ implica che $d > 1$. Sia $j = m/d$, che è un intero diverso da zero in $\mathbb{Z}_m$ perché $d > 1$. Allora la seguente mostra che $h^j = 1$, completando la dimostrazione:

$$h^j = g^{ij} = g^{i \cdot m/d} = g^{m \cdot i/d} = (g^m)^{i/d} = 1^{i/d} = 1$$

Esempio:

Determiniamo tutti i generatori del gruppo $\mathbb{Z}_{11}^*$. Usiamo prima la Proposizione 1.21. La dimensione di $\mathbb{Z}_{11}^*$ è $m = \phi(11) = 10$ e la scomposizione in fattori primi di 10 è $2 \cdot 5$. Quindi, il test per stabilire se un dato $a \in \mathbb{Z}_{11}^*$ è un generatore è che $a^2 \not\equiv 1 \mod 11$ e $a^5 \not\equiv 1 \mod 11$. Calcoliamo $a^2 \mod 11$ e $a^5 \mod 11$ per tutti gli elementi del gruppo a.

a	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
$a^2 \mod 11$	1	4	9	5	3	3	5	9	4	1
$a^5 \mod 11$	1	10	1	1	1	10	10	10	1	10

I generatori sono quelle a per cui la colonna corrispondente non ha una voce uguale a 1, il che significa che in entrambe le righe la voce per questa colonna è diversa da 1. Quindi:

$$Gen(\mathbb{Z}_{11}^*) = \{2, 6, 7, 8\}$$

Ora, usiamo la Proposizione 1.20 e ricontrolliamo che otteniamo la stessa cosa. Abbiamo visto nell'Esempio precedente che 2 era un generatore di $\mathbb{Z}_{11}^*$. Secondo la Proposizione 1.20, l'insieme dei generatori è:

$$Gen(\mathbb{Z}_{11}^* = \{2^i \mod 11 : i \in \mathbb{Z}_{10}^*\})$$

Questo perché la dimensione del gruppo è $m = 10$. Ora, $\mathbb{Z}_{10}^* = \{1, 3, 7, 9\}$. I valori di $2^i \mod 11$ come i varia su questo insieme possono essere ottenuti dalla tabella dove abbiamo calcolato tutte le potenze di 2. Quindi:

$$\{2^i \mod 11 : i \in \mathbb{Z}^*_{10}\} = \{2^1 \mod 11, 2^3 \mod 11, 2^7 \mod 11, 2^9 \mod 11\} =$$
$$= \{2, 6, 7, 8\}$$

Questo è lo stesso insieme che abbiamo ottenuto sopra tramite la Proposizione 1.19. Se proviamo a trovare un generatore selezionando casualmente gli elementi del gruppo e poi testando usando la Proposizione 1.19, ogni prova ha probabilità di successo $\phi(10)/10 = 4/10$, quindi ci aspetteremmo di trovare un generatore in $10/4$ prove. Possiamo ottimizzare leggermente notando che 1 e $p-1$ non possono mai essere generatori, e quindi abbiamo solo bisogno di scegliere casualmente i candidati da $\mathbb{Z}^*_{11} - \{1, 10\}$. In tal caso, ogni prova ha probabilità di successo $\phi(10)/8 = 4/8 = 1/2$, quindi ci aspetteremmo di trovare un generatore in 2 prove.

Quando si vuole lavorare in un gruppo ciclico in crittografia, la scelta più comune è lavorare su $\mathbb{Z}^*_p$ per un opportuno primo p. L'algoritmo per trovare un generatore consiste nel ripetere il processo di selezione di un elemento del gruppo casuale e di testarlo, fermandosi quando viene trovato un generatore. Per renderlo possibile scegliamo p in modo tale che sia nota la scomposizione in fattori primi dell'ordine $p-1$ di $\mathbb{Z}^*_p$. Per rendere il test veloce, scegliamo p in modo che $p-1$ abbia pochi fattori primi.
Di conseguenza, è comune scegliere p uguale a $2q+1$ per qualche primo q. In questo caso, la scomposizione in fattori primi di $p-1$ è $2 \cdot q$, quindi dobbiamo elevare un candidato a solo due potenze per verificare se è un generatore o meno. Nella scelta dei candidati, ottimizziamo leggermente notando che 1 e $p-1$ non sono mai generatori, e di conseguenza scegliamo i candidati da $\mathbb{Z}^*_p - \{1, p-1\}$ piuttosto che da $\mathbb{Z}^*_p$. L'algoritmo è il seguente:

```
FIND-GEN(p):
    q <- (p - 1)/2
    found <- 0
    while (found != 0) do:
        g <-R Z*_p - {1, p - 1}
        if (g^2 mod p != 1 and g^q mod p != 1) then found <- 1
    return g
```

Quadrati e non quadrati

Un elemento a di un gruppo G è detto *quadrato*, o *quadrato residuo* se ha la radice quadrata, cioè esiste un $b \in G$ tale che $b^2 = a \in G$. Diciamo che

$$QR(G) = \{g \in G : \exists b \in G \land g = b^2\}$$

indica l'insieme di tutti i quadrati del gruppo G.

Ci interessa soprattutto il caso in cui il gruppo G sia $\mathbb{Z}^*_N$ per qualche intero N. Un intero a è chiamato **quadrato mod N** o **residuo quadratico mod N** se $a \mod N$ è un membro di $QR(\mathbb{Z}^*_N)$. Se $b^2 \equiv a \mod N$ allora b è detto radice quadrata di $a \mod N$. Un intero a è chiamato **non quadrato mod N** o **quadratico non residuo mod N** se $a \mod N$ è un membro di $\mathbb{Z}^*_N - QR(\mathbb{Z}^*_N)$. Inizieremo esaminando il caso in cui $N = p$ è primo. In questo caso definiamo una funzione $J_p : \mathbb{Z} \to \{-1, 0, 1\}$

$$J_p(a) = \begin{cases} 1 \text{ se } a \text{ è un quadrato residuo mod p} \\ 0 \text{ se } a \; mod \; p = 0 \\ -1 \text{ altrimenti} \end{cases}$$

per tutte le $a \in \mathbb{Z}$. Chiamiamo $J_p(a)$ il **simbolo di Legendre** di a. Quindi, il simbolo di Legendre è semplicemente una notazione compatta per dirci se il suo argomento è o meno un quadrato modulo p.
Prima di passare allo sviluppo della teoria, può essere utile guardare un esempio.

Proposizione 1.21 Sia $p > 2$ un numero primo e sia g un generatore di $\mathbb{Z}_p^*$. Allora

$$QR(\mathbb{Z}_p^*) = \{g^i : i \in \mathbb{Z}_{p-1} \wedge i \mod 2 = 0\}$$

e il numero di quadrati mod p è:

$$\left|QR(\mathbb{Z}_p^*)\right| = \frac{p-1}{2}$$

Inoltre, ogni quadrato mod p ha esattamente due diverse radici quadrate mod p.

Proposizione 1.22 Sia $p > 2$ un numero primo. Allora:

$$J_p(a) \equiv a^{\frac{p-1}{2}} \mod p$$

per ogni $a \in \mathbb{Z}_p^*$.

Lemma 1.23 Sia $p > 2$ un numero primo. Allora:

$$g^{\frac{p-1}{2}} \equiv -1 \mod 11$$

per ogni generatore g di $\mathbb{Z}_p^*$.

Esempio:

Consideriamo $\mathbb{Z}_{11}^*$, per cui sappiamo che $Gen(\mathbb{Z}_{11}^*) = \{2, 6, 7, 8\}$.

Allora per il Lemma 1.23 si ha:

$$2^5 \equiv -1 \mod 11$$
$$2^5 \mod 11 = -1 \mod 11$$
$$32 \mod 11 = -1 \mod 11$$
$$32 - 3 * 11 = -1$$
$$-1 = -1$$

Si può provare che il Lemma vale anche per gli altri generatori.

Dimostrazione Proposizione 1.22

Per definizione del simbolo di Legendre, dobbiamo mostrare che:

$$a^{\frac{p-1}{2}} \equiv \begin{cases} 1 \mod p \text{ se } a \text{ è un quadrato mod p} \\ -1 \mod p \text{ altrimenti} \end{cases}$$

Sia g un generatore di $\mathbb{Z}_p^*$ e sia $i = DLog_{\mathbb{Z}_p^*, g}(a)$. Consideriamo separatamente i casi in cui a è un quadrato residuo e a non è un quadrato residuo.

Supponiamo che a sia un quadrato residuo. Allora per la Proposizione 1.21 i è pari. In questo caso:

$$a^{\frac{p-1}{2}} \equiv (g^i)^{\frac{p-1}{2}} \equiv (g^{p-1})^{i/2} \equiv 1 \mod p$$

Adesso supponiamo che a non sia un quadrato residuo. Allora per la Proposizione 1.21 i è dispari. In questo caso:

$$a^{\frac{p-1}{2}} \equiv (g^i)^{\frac{p-1}{2}} \equiv g^{(i-1)\cdot\frac{p-1}{2}+\frac{p-1}{2}} \equiv (g^{p-1})^{(i-1)/2} \cdot g^{\frac{p-1}{2}} \equiv g^{\frac{p-1}{2}} \mod p$$

Il Lemma 1.23 ci dice che l'ultima quantità è -1 modulo p, come desiderato.

Curve Ellittiche

Le Curve Ellittiche (ed iperellittiche) su campi finiti sono molto utili in crittografia in particolare per:

- fattorizzare;
- test di primalità;
- schemi di cifratura.

Da tali curve si possono ricavare un numero enorme di gruppi abeliani, "algebricamente" molto ricchi.

La matematica delle Curve Ellittiche è estremamente complessa, pertanto non verrà trattata e dove possibile si accennerà qualcosa.