Cryptology

February 29, 2016

Contents

1 Modular arithmetic

1.1 Prerequisites: basic properties of the integers

Definition: Divisibility

Let a and b two integers. Then a divides b (denoted as a|b) if there exists an integer c such that b = c * a.
In this case, we also say that b is a multiple of a

Property

- $\forall c \in \mathbb{Z}$, 1|c and c|c (reflexivity)
- If a|b and b|c then a|c (transitivity)
- If a|b and a|c then a|(b+c)
- $\forall c$ integer $c \neq 0$ a|b $\Leftrightarrow$ ac|bc

Definition 1.1. Prime numbers A prime number is a positive integer $p \neq 1$ that is only divisible by ± 1 and p.
The set of prime numbers is denoted by P:
$P = \{2, 3, 5, 7, 11, ...\}$
A positive integer that is not a prime number is called a composite.

Theorem 1.1.
There are infinitely many prime numbers.
Let $\Pi(n)$ *the number of prime numbers smaller than* n*.*
Then $\Pi(n) \underset{n\to\infty}{\sim} \frac{n}{log(n)}$

Remark

The probability that a random integer n *is prime is about* $\frac{1}{log(n)}$

Theorem 1.2 (Fundamental theorem of arithmetic)**.**
Every non zero integer n *can be written as a product of primes*:
$n = \pm 1 * p_1^{\alpha_1} * p_2^{\alpha_2} ... * p_k^{\alpha_k}$*, where* $p_i \in P$*, and* $\alpha_i \in N$
This decomposition is unique if $p_1 < p_2 < ... < p_k$ *and* $\alpha > 0$ $\forall i$

Proof. The proof uses Euclid's lemma.

- **Existence**
  We need to show that every integer greater than 1 is a product of primes.
  By induction: assume it is true for all numbers between 1 and n.
  If n is prime, there is nothing more to prove (a prime is a trivial product of primes, a "product" with only one factor). Otherwise, there are integers

a and b, where n = ab and $1 < a \leq b < n$. By the induction hypothesis, a = $p_1 p_2 ... p_j$ and b = $q_1 q_2 ... q_k$ are products of primes. But then n = ab = $p_1 p_2 ... p_j q_1 q_2 ... q_k$ is a product of primes.

- **Uniqueness**
  Assume that $s > 1$ is the product of prime numbers in two different ways:

$$\begin{aligned} s &= \mathrm{p}_1 p_2 ... p_m \\ &= \mathrm{q}_1 q_2 ... q_n \end{aligned}$$

We must show m = n and that the q_j are a rearrangement of the p_i.
By Euclid's lemma, p_1 must divide one of the q_j; relabeling the q_j if necessary, say that p_1 divides q_1. But q_1 is prime, so its only divisors are itself and 1. Therefore, $p_1 = q_1$, so that

$$\begin{aligned} s/\mathrm{p}_1 &= \mathrm{p}_2 ... p_m \\ &= \mathrm{q}_2 ... q_n \end{aligned}$$

Reasoning the same way, p_2 must equal one of the remaining q_j. Relabeling again if necessary, say $p_2 = q_2$. Then

$$\begin{aligned} \mathrm{s}/(\mathrm{p}_1 p_2) &= \mathrm{p}_2 ... p_m \\ &= \mathrm{q}_3 ... q_n \end{aligned}$$

This can be done for each of the m p_i's, showing that $m \leq n$ and every p_i is a q_j. Applying the same argument with the p's and q's reversed shows $n \leq m$ (hence m = n) and every q_j is a p_i.

□

Lemma 1.3 (Euclid's lemma)**.** *Let p a prime number and $a, b \in \mathbb{Z}$. Then $p/ab \Rightarrow p/a$ or p/b.*

Proof. The usual proof involves another lemma called Bezout's identity. This states that if x and y are relatively prime integers (i.e. they share no common divisors other than 1) there exist integers r and s such that

$$rx + sy = 1.$$

Let a and n be relatively prime, and assume that n|ab. By Bezout's identity, there are r and s making

$$rn + sa = 1.$$

Multiply both sides by b:

$$rnb + sab = b.$$

The first term on the left is divisible by n, and the second term is divisible by ab which by hypothesis is divisible by n. Therefore their sum, b, is also divisible by n. This is the generalization of Euclid's lemma mentioned above. □

Asymptotic notations and complexity basics

f, g real functions, g is positive:

- $f = O(g)$ *if there exists a constant* $c > 0$ *such that* $|f(x)| \leq c * g(x)$ *for any x sufficiently large.*
- $f = o(g)$ *if* $\frac{f}{g}(x) \underset{n\to\infty}{\to} 0$
- $f \sim g$ *if* $\frac{f}{g}(x) \underset{n\to\infty}{\to} 1$
- $f = \Theta(g)$ *if there exists* c_1, c_2 *such that* $c_1 * g(x) \leq |f(x)| \leq c_2 * g(x)$
- $f = \Omega(g)$ *if there exists a constant c such that* $f(x) \geq c * g(x)$ *for x sufficiently large*

Property

- $f = o(g) \Rightarrow f = O(g),\ g \neq O(f)$
- $f \sim g \Leftrightarrow f = (1 + o(1))\ g$

The size of an integer a is the number of bits in the binary representation of $|a|$, *that is* $\lfloor log_2(|a|) \rfloor + 1$

Polynomial time algorithm

Algorithm whose running time is bounded by a polynomial in the length of the input. i.e. the complexity is in $exp^{O(1)*log(n)}$, *where n is the size of the input*

Exponential time algorithm

Algorithm whose running time is exponential in the length of the input.
i.e. the complexity is in $exp^{O(1)*n}$

Sub-exponential time algorithm

Complexity is "in between" poly and exponential complexities. More precisely the complexity is in:

$L_n = exp(O(1) * n^{\alpha} * (log(n))^{1-\alpha})$ *where* $0 < \alpha < 1$

- $\alpha = 0$ *poly complexity*
- $\alpha = 1$ *expo complexity*

where n is the size of the input

1.2 Congruences

Theorem 1.4 (Euclidean division).
For a, b, c $\in \mathbb{Z}$, $b \neq 0$, there exist a unique q (quotient), r (remainder) $\in \mathbb{Z}$ such that

- $a = b * q + r$
- $0 \leq r < |b|$

Definition 1.2. Congruence Let x, y, n $\in \mathbb{Z}$. Then x is congruent to y modulo n if their remainders in the division by n are the same.
In particular

$$x = y mod n \Leftrightarrow \text{n | (x-y)}$$
$$\Leftrightarrow \exists k \in Z, x = k * n + y$$

Property

- *Thus x an equivalence relation (reflexive, transitive and symmetric)*

- *Compatibility with addition and multiplication modulo n*
  $\forall a, b, a', b' \in \mathbb{Z}$ such that a = a' mod n and b = b' mod n. Then
  - *a+b = a' + b' mod n*
  - *a*b = a' * b' mod n*

  The congruence relation partition Z into equivalent classes:

Definition 1.3. Residue classes mod m

- $\mathbb{Z}/n\mathbb{Z}$ is the set of equivalence (or residue)
- For any integer m in a residue class, we call m a representative of that class

Note: there are precisely n distinct residue classes modulo n, given for example by $\bar{0}$, $\bar{1}$,$\bar{2}$, ..., $\overline{n-1}$

Property

*($\mathbb{Z}/n\mathbb{Z}$, +, *) is a ring*

1.3 Modular exponentiation

Question: Given $x \in \mathbb{Z}/n\mathbb{Z}$ and $e \in N^*$. How to compute x^e mod n ?

- *naive approach*

$$x^2 \ = x * xmodn$$
$$x^3 \ =x^2 \ * \ x \ \ mod \ \ n$$
$$...$$
$$x^e = x^{e-1} * x \ mod \ n$$

e multiplications, each one is of cost $O((log_2(n))^2) \Rightarrow O(e*(log_2(n))^2)$

- *second approach*

$$e = \sum_{i=0} le_i * 2^i \ where \ e_i \in 0,1$$
$$x^e = x^{\sum_{i=0} le_i * 2^i}$$
$$= \prod_{i=0} l(x^{2^i})^{e_i}$$

Example:

$$x^{1024}[n] = x^{2^{10}}$$
$$x_2 = x^2[n]$$
$$x_4 = x_2^2 = x^4[n]$$
$$x_8 = x_4^2 = x^8[n]$$
$$x_{16} = x^{16}[n]$$
$$...$$
$$10 sequences in total$$

Property

Let $e = (e_l ... e_0)_2$ *the binary expression of e,*
i.e. $e = \sum_{i=0}^{l-1} e_i * 2^i$.
Then

$$x^e = \prod_{i=0}^{l-1} (x^{2^i} \ mod \ n)^{e_i}$$
$$= \prod_{i=0, e_i \neq 0}^{l-1} (x^{2^i}) \ mod \ n$$

Algorithm: "Right to left" modular exponentiation

Input: $x \in \mathbb{Z}/n\mathbb{Z}$, $n \in N^*$, $e \in N^*$
Output: $y = x^e \ mod \ n$
$y \leftarrow 1$
$t \leftarrow x \ mod \ n$
while *e != 0* ***do***
  if *e = 1 mod 2* ***then***
    $y \leftarrow$ *y * t mod n*
  end if
  $t \leftarrow t^2 \ mod \ n$
  $e \leftarrow$ *e » 1*

end while
return y

Remark

The complexity is in O(log(e (log n) ^{2}*))*$\rightarrow$ *Polynomial algorithm*
Given $n \in \mathbb{N}*, x \in \mathbb{Z}/n\mathbb{Z}$ *and e, it is easy to compute* x^e *mod n.*
However, there is no efficient (polynomial) algorithm which computes e given x^e*, n, x*
$\rightarrow$ *this is called the discrete logarithm problem.*

1.4 Extended Euclid algorithm

Definition 1.4. GCD, LCM, coprimality For $a, b \in \mathbb{Z}$, we call gcd(a, b) or $a \wedge b$ the greatest common divisor of a and b and lcm(a, b) or $a \vee b$ their least common multiple.

In particular:

- x | a and x | b ⇒ x | $(a \wedge b)$
- a | m and b | m ⇒ (a V b) | m

Property

If:

- *a =*
- *b =*

where blablabla
then $a \wedge b$ *= iets*
$a \vee b$ *= iets*
In particular $(a \wedge b)$***$(a \vee b)$ *= a*b*

Lemma 1.5 (Property: Gaus Lemma)**.** *If p,q coprime and* $x \in \mathbb{Z}$ *such that p/q*x*
Then p/x

Lemma 1.6 (Bezout)**.** *For a, b, c* $\in \mathbb{Z}$, $\exists u, v \in Z$ *such that* $u * a + v * b = gcd(a, b)$.

Proof. If r is the remainder in the division of a by b: a = q*b + r ($0 \leq r < |b|$). Then $a \wedge b = b \wedge r$. (*)

Now let $r_0 = a$, $r_1 = b$. We compute the iteratively:

$$\begin{aligned}
\mathrm{r}_0 &= \mathrm{r}_1 * q_1 + r_2 \text{ where } 0 \leq r_2 < |r_1| \\
\mathrm{r}_1 &= \mathrm{r}_2 * q_2 + r_3 \text{ where } 0 \leq r_3 < |r_2| \\
&\dots \\
\mathrm{r}_{n-2} &= \mathrm{r}_n - 1 * q_{n-1} + r_n \text{ where } 0 \leq r_n < |r_{n-1}| \\
\mathrm{r}_{n-1} &= \mathrm{r}_n * q_n + r_{n+1} \text{ where } r_{n+1} = 0
\end{aligned}$$

(*) Thus $r_0 \wedge r_1 = r_1 \wedge r_2 = ... = r_{n-1} \wedge r_n = r_n \wedge r_{n+1} = r_n$

At the end, we have $a \wedge b$ is equal to the last non zero remainder r_n.
Goal: u, v such that $a * u + b * v = a \wedge b$.
We define (u_i) and (v_i) such that

$$\begin{cases} u_0 = 1 \text{ and } v_0 = 0 \\ u_1 = 0 \text{ and } v_1 = 1 \end{cases}$$

$[u_i$ * a + v_i * b = $r_i]$
u_0 * a + v_0 * b = r_0 = a
u_1 * a + v_1 * b = r_1 = b

□

Induction Hypothesis

$$\begin{cases} u_{i-1} *a + v_{i-1} *b = r_{i-1} \\ u_i *a + v_i *b = r_i \end{cases}$$

$$\begin{aligned} r_{i+1} &= r_i * q_i - r_{i-1} \\ &= (u_i * a + v_i * b) * q_i - (u_{i-1} * a + v_{i-1} * b) \\ &= (u_i * q_i - u_{i-1}) * a + (v_i * q_i - v_{i-1}) * b \\ &= u_{i+1} * a + v_{i+1} * b \end{aligned}$$

Corresponding pseudo-code

Euclidian algorithm
Input: a, b integers
Output: $a \wedge b$
$r_0 \leftarrow a$
$r_1 \leftarrow b$ *mod* n
while $r_1 != 0$ ***do***
temp $\leftarrow r_0$
$r_0 \leftarrow r_1$
$r_1 \leftarrow tmp\%r_0$ *(remainder in div of initial by* r_i*)*
end while
return r_0

Euclidian Extended algorithm
Input: a, b integers
*Output: u, v such that u*a + v*b = gcd(a, b)*
$u_0 \leftarrow 1$
$u_1 \leftarrow 0$
while *b != 0* ***do***
temp $\leftarrow a$
$a \leftarrow b$
$b \leftarrow tmp\%a$
$q \leftarrow tmp\%a$
$tmp \leftarrow u_0 - q * u_1,\ u_0 \leftarrow u_1, u_1 \leftarrow tmp$
$tmp \leftarrow v_0 - q * v_1,\ v_0 \leftarrow v_1, v_1 \leftarrow tmp$

end while
return u_0, v_0

Theorem 1.7 (Chinese remainder theorem (CRT))**.** *Let m, n co-prime integers. Let a and b be two integers. Then the system (S)* $\begin{cases} x = a \bmod n \\ x = b \bmod m \end{cases}$ *admits a unique solution modulo m*n*

Proof. Bezout $\Rightarrow \exists u, v$ such that $u * m + v * n = 1$.
Consider $x_0 = a * u * m + b * v * n$.
Then

$$\begin{aligned} x_0 &= a*u*m \text{ mod n} \\ &\overset{Bezout}{=} a * 1 mod n \\ &= a \text{ mod n} \end{aligned}$$

And

$$\begin{aligned} x_0 &= b*v*n \text{ mod n} \\ &\overset{Bezout}{=} b * 1 mod n \\ &= b \text{ mod n} \end{aligned}$$

$\Rightarrow x_0$ is a solution of the system

- If x_1 is an other solution of (S) modulo m*n,

then $\begin{cases} x_0 = x_1 mod n \\ x_0 = x_1 mod m \end{cases} \Leftrightarrow \begin{cases} n \mid (x_0 - x_1) \\ m \mid (x_0 - x_1) \end{cases}$
$\Leftrightarrow m * n | (x_0 - x_1) \Leftrightarrow x_0 = x_1$ mod m*n

□

1.5 Modular inverse

Definition 1.5. Let x, m $>$ 0 be integers. We say x is invertible mod n if there exists $y \in \mathbb{Z}$ such that x*y $=$ 1 mod n.
This is denoted $x^{-1} =$ y mod n
Similarly, $x \in \mathbb{Z}/n\mathbb{Z}$ is invertible if $\exists y \in \mathbb{Z}/n\mathbb{Z}$, x*y $=$ 1 mod n

Theorem 1.8. *An integer a is invertible modulo n if and only if* $a \wedge n = 1$

Proof. The proof goes as follows:

$\Leftarrow$: $a \wedge n = 1$
$\exists u, v \in \mathbb{Z}$, $a * u + b * v = 1$
$u = a^{-1} mod n$

$\Rightarrow$: $\exists y \in \mathbb{Z}/n\mathbb{Z}$, a*y $=$ 1 mod n
$\Rightarrow$, $\exists k : a * y + k * n = 1$
$\Rightarrow a \wedge n = 1$

□

Remark: p prime

$a \in (\mathbb{Z}/p\mathbb{Z})^* = \mathbb{Z}/p\mathbb{Z}\ 0$
$\Rightarrow$ *a invertible*

Euclidean Extended algorithm

Input: $a,\ n \in \mathbb{Z}$
Output: a^{-1} *mod* n
$u_0 \leftarrow 1$
$u_1 \leftarrow 0$
while n *!= 0* ***do***
  $temp \leftarrow a$
  $a \leftarrow n$
  $n \leftarrow tmp\%a$
  $q \leftarrow tmp\%a$
  $tmp \leftarrow u_0 - q * u_1$
  $u_0 \leftarrow u_1, u_1 \leftarrow tmp$
end while
return u_0

Definition 1.6. Euler totient function It is defined by
$\forall n \in N^*, \Phi(n) = \#(\mathbb{Z}/n\mathbb{Z})^*$, where $\mathbb{Z}/n\mathbb{Z}$ is the set of invertible elements in $\mathbb{Z}/n\mathbb{Z}$.

Examples:

$\Phi(1) = 1$

$\Phi(2) = 1$

$\Phi(3) = 2$

$\Phi(4) = 2$

Property: computation of Euler's totient function

(i) $\Phi(m * n) = \Phi(m) * \Phi(n)$ *if* $m \wedge n = 1$
(ii) $\Phi(p^e) = p^e - p^{e-1} = p^e * (1 - \frac{1}{p})$ *if* p *is prime and* $e > 0$
(iii) $\Phi(n) = n * \prod_{i=1}^{k} (1 - \frac{1}{p_i})$ *where* $n = p_1^{\alpha_1} * p_2^{\alpha_2} * ... * p_k^{\alpha_k}$ *is the factorization of n.*

Proof. The proof goes as follows:

(ii) p prime
$(\mathbb{Z}/p\mathbb{Z})^* = (\mathbb{Z}/p\mathbb{Z})^* \Rightarrow \Phi(p) = p - 1$
$e \in N^*\ p^e \wedge x = 1$
$0 \leq$ x $< p^e$
x $\wedge p^e \neq 1$
Or x $\wedge p^e =$ p
$0 \leq$ k*p $< p^e \rightarrow$p^{e-1} choices for k

(i) CRT
m $\wedge$ n $= 1$
$\mathbb{Z}/n\mathbb{Z}$ * $\mathbb{Z}/m\mathbb{Z} \xrightarrow{\sim} \mathbb{Z}/mn\mathbb{Z}$: this is a bijection
$(\mathbb{Z}/n\mathbb{Z})^*$ * $(\mathbb{Z}/m\mathbb{Z})^* \xrightarrow{\sim} (\mathbb{Z}/mn\mathbb{Z})^*$

$n = p_1^{\alpha_1} * p_2^{\alpha_2} * ... * p_k^{\alpha_k}$, where p_i different prime and n > 0.

$$\begin{aligned}\Phi(n) &= \Phi(p_1^{\alpha_1}) * \Phi(p_2^{\alpha_2}) * ... * \Phi(p_k^{\alpha_k}) \\ &= \mathrm{p}_1^{\alpha_1} * (1\text{-}\tfrac{1}{p_1}) * ... * p_k^{\alpha_k} * (1\text{-}\tfrac{1}{p_k}) \\ &= \mathrm{p}_1^{\alpha_1} * ... * p_k^{\alpha_k} * \prod_{i=1}^{k} (1 - \tfrac{1}{p_i})\end{aligned}$$

$\square$