CSCI 3270 Advanced Programming Laboratory
Number Theory

MW

5 March 2014

1 Euclidean Algorithm

> 又有九十一分之四十九。問約之得幾何？
> 答曰：十三分之七。
> 約分術曰：可半者半之，不可半者，副置分母子之數，以少減多，更相減損，求其等也。以等數約之。　《九章算術・方田》

Notation Denote (a, b) as Greatest Common Divisor(GCD) of a and b.

Without loss of generality, assume $a > b$. Euclidean Algorithm gives the following.

Definition We say that a and b co-prime to each other if and only if $(a, b) = 1$

Theorem 1.1 $(a, b) = (a \bmod b, b)$

Proof. First, we prove a weaker statment: $(a, b) = (a - b, b)$

Assume $g = (a, b)$. Let $a = a' \times g$ and $b = b' \times g$ where a' and b' co-prime to each other.

$$(a - b, b) = ((a' - b')g, b'g)$$

It can be proved that $(a' - b')$ and b' co-prime. The proof is so easy that leave it to readers. Therefore, $(a - b, b)$ also equals to g.

By induction, $(a, b) = (a \bmod b, b)$.

Theorem 1.2 For all integer a and b, there exist integers x and y such that $ax + by = (a, b)$.

When $b = 0$ the question is trivial: $x = 1$ and $y = 0$
Otherwise, let $bx' + (a \bmod b)y' = (a, b)$ and $a = \lfloor \frac{a}{b} \rfloor b + a \bmod b$

$$\begin{array}{rcl|l} ax + by & = & bx' + (a \bmod b)y' & \\ (\lfloor \frac{a}{b} \rfloor b + a \bmod b) \times y' + by & = & bx' + (a \bmod b)y' & \text{Let x = y'} \\ \lfloor \frac{a}{b} \rfloor by' + by & = & bx' & \\ x = y' \quad \text{and} & & y = x' - \lfloor \frac{a}{b} \rfloor y' & \end{array}$$

Pseudocode Extended Euclidean Algorithm

```
1: function EGCD(Integer a, Integer b)
2:     if b = 0 then
3:         return x = 1, y = 0
4:     else
5:         y, x ← EGCD(b, a mod b)
6:         return x, y − ⌊a/b⌋x
7:     end if
8: end function
```

Example Problems POJ 1061 青蛙的約會

Theorem 1.3 For all linear combination of a and b

$$gcd(a, b) \mid xa + yb$$

The proof is so easy. Do it yourself.

Applying division under modulate is different. When dividing a number by k, we instated multiply the multiplicative inverse of k. Denote as $inv(k)$.

$$\begin{aligned} k \times inv(k) &= 1 \pmod{n} \\ k \times inv(k) + jn &= 1 \end{aligned}$$

This leads to the following theorem.

Theorem 1.4 There exist $inv(k)$ of an integer k under modulate $n \Leftrightarrow (k, n) = 1$

2 Method of Successive Substitution

> 今有物，不知其數。三、三數之，賸二；五、五數之，賸三；七、七數之，賸二。問物幾何？
> 答曰：二十三。 《孫子算經・卷下》

Notation We denote $a \equiv b \pmod{p}$ as a equivalent to b under mod p. i.e. there exist an integer k such that $a + kp = b$.

Example

$$\begin{aligned} -5 \equiv 0 \equiv 5 \equiv 10 \equiv \ldots \pmod 5 \\ -4 \equiv 1 \equiv 6 \equiv 11 \equiv \ldots \pmod 5 \end{aligned}$$

In this section, we are going to solve the following system of equations.

$$\begin{cases} x \equiv a_1 \pmod{n_1} \\ x \equiv a_2 \pmod{n_2} \\ \vdots \\ x \equiv a_k \pmod{n_k} \end{cases}$$

Here, we first consider an easier case. Where there are only two equations, and n_1 and n_2 co-prime to each other.

Theorem 2.1 Chinese Remainder Theorem gives that in such a case, x can be uniquely determined under modulate $n_1 \times n_2$.

$$\begin{cases} x \equiv a_1 \pmod{n_1} \Rightarrow x = k_1 n_1 + a_1 \\ x \equiv a_2 \pmod{n_2} \Rightarrow x = k_2 n_2 + a_2 \end{cases}$$

$$\begin{aligned} k_1 n_1 + a_1 &= k_2 n_2 + a_2 \\ k_1 n_1 - k_2 n_2 &= a_2 - a_1 \end{aligned}$$

By Theorem 1.2, solve k_1, k_2 of the above equation using Extended Euclidean Algorithm and it is done.

After solving system of equations with two equations, by induction, system with more equations can be solved.

We now move on to the case that $(n_1, n_2) \neq 1$. There are several ways to solve this.

Method 1 Split equations which n_k is not a power of prime into several equations modulate prime powers.
For example:

$$x = 72 \pmod{140} \Rightarrow \begin{cases} x = 0 \pmod 4 \\ x = 2 \pmod 5 \\ x = 2 \pmod 7 \end{cases}$$

Then, contradiction between equations can be found easily, so as the solution.

Method 2 Method of Successive Substitution
Given two equations

$$\begin{cases} x \equiv a_1 \pmod{n_1} \quad ... \quad (i) \\ x \equiv a_2 \pmod{n_2} \quad ... \quad (ii) \end{cases}$$

Without loss of generality, assume $n_1 \leq n_2$ and n_1 and n_2 no necessary to be co-prime.

Consider (i), $x \equiv a_1 \pmod{n_1} \Rightarrow x = a_1 + jn_1$ with j as an integer.
Substitute this into (ii) we get $a_1 + jn_1 \equiv a_2 \pmod{n_2} \Rightarrow jn_1 \equiv a_2 - a_1 \pmod{m}$ ... (iii). Now, denote $n_1' = \dfrac{n_1}{(n_1, n_2)}$ and $n_2' = \dfrac{n_2}{(n_1, n_2)}$.
From (iii) we obtain $jn_1' = \dfrac{a_2 - a_1}{(n_1, n_2)} \pmod{n_2'} \Rightarrow j = \dfrac{a_2 - a_1}{(n_1, n_2)} \times inv_{n_2'}(n_1') \pmod{n_2'} \Rightarrow j = \dfrac{a_2 - a_1}{(n_1, n_2)} \times inv_{n_2'}(n_1') + in_2'$.

Here, there is an integer solution j if and only if $(n_1, n_2) | a_2 - a_1$. Otherwise, this equation have no solution.
By Theorem 1.3 there must exist an $inv_{n_2'}(n_1')$ since n_1' and n_2' co-prime.

Substitute this into first equation, we get

$$\begin{aligned} x &= a_1 + (\frac{a_2 - a_1}{(n_1, n_2)} inv_{n_2'}(n_1') + in_2')n_1 \\ &= a_1 + \frac{a_2 - a_1}{(n_1, n_2)} n_1 \times inv_{n_2'}(n_1') + in_1 n_2' \\ &= a_1 + \frac{a_2 - a_1}{(n_1, n_2)} n_1 \times inv_{n_2'}(n_1') \qquad (\text{mod } \frac{n_1 n_2}{(n_1, n_2)}) \\ &= a_1 + \frac{a_2 - a_1}{(n_1, n_2)} n_1 \times inv_{n_2'}(n_1') \qquad (\text{mod } LCM(n_1, n_2)) \end{aligned}$$

3 Discrete Logarithm

小知不及大知，小年不及大年。奚以知其然也？ 《莊子・逍遙遊》

$$a^b = c \pmod{P}$$

This form of equations appear frequently. Usually, P and two of a, b and c are given. Aim to find the remaining one.

Problem Given a and b, aim to find c, a^b.
This problem is trivial. Using divide and conquer, c can be computed with $\Theta(\log b)$.

Pseudocode Computing $a^b \mod P$

```
1: function POWER(Integer a, Integer b)
2:     if b = 0 then
3:         return 1
4:     else if b is odd then
5:         return POWER(a × a mod P, ⌊b/2⌋) × a mod P
6:     else if b is even then
7:         return POWER(a × a mod P, ⌊b/2⌋)
8:     end if
9: end function
```

Problem Given b and c, aim to find a, $\sqrt[b]{c}$
As P is a prime, $c^{\varphi(P)} = c^{P-1} = 1 \pmod{P}$ (Read part 6 for more information of Euler Phi Function and Fermat Little Theorem)

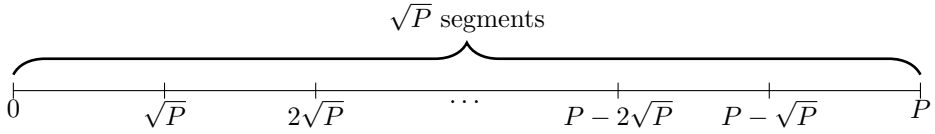
$$a = \sqrt[b]{c} = \sqrt[b]{c^{k(P-1)+1}} \pmod{P}$$

If there is a non-negative integer k satisfying $b \mid k(P-1)+1$, it will be a solution. By rewritting the condition as $jb = k(P-1)+1$, solutions can be computed by Extended Euclidean Algorithm. (Note that sometime there are no solution)

Problem Given a and c, aim to find b, $\log_a c$.
The problem is called Discrete Logarithm.

In continue case, the exponent can be computed by $b = \log_a c$. In discrete case, here we introduce a method to compute exponent with time complexity $\Theta(\sqrt{P} \log P)$.

This algorithm is called Baby-step Giant-step Algorithm. The idea is that, as $0 \leq b \leq \varphi(P) < P$, this range can be divided into $\sqrt{P}$ parts, each part with $\sqrt{P}$ numbers. See the figure below.



The equation rewritten as follow

$$\begin{array}{rcll|l} a^{i\sqrt{P}+j} & = & c & \pmod P & 0 \leq i, j \leq \sqrt{P} \\ a^j & = & \dfrac{c}{a^{i\sqrt{P}}} & \pmod P & \\ a^j & = & c \times inv(a^{i\sqrt{P}}) & \pmod P & inv() \text{ as multiplicative inverse function} \end{array}$$

The answer will be search as the following way.

1. Pre-compute a^j with $0 \leq j \leq \sqrt{P}$
2. Exhaust every $a^{i\sqrt{P}}$ with $0 \leq i \leq \sqrt{P}$
3. Compute $c \times inv(a^{i\sqrt{P}})$
4. Search if the number correspond to any a^j
5. If yes, $b = i\sqrt{P} + j$
   If no, comtinue searching

Pseudocode Baby-step Giant-step Algorithm

```
P' ← ⌈√P⌉
array arr ← {a^0, a^1, ..., a^P'}                ▷ Pre-compute Baby-Step
sort arr
for i from 1 to P' do
    if c × inv(a^(iP')) ∈ arr then               ▷ Giant-step, binary search Baby-step
        Return the answer
    end if
end for
```

4 Lucas' Theorem

聖人見微以知萌，見端以知末，故見象箸而怖，知天下不足也。
《韓非子・説林上》

It is a hard problem to calculate binomial coefficient when parameter growth, since there are factorial calculation in the defination. Édouard Lucas gives a method to calculate binomial coefficient under modulating a relatively small prime p.

$$\binom{m}{n} = \prod_{i=0}^{k} \binom{m_i}{n_i} \pmod p$$

Here, $m = m_k p^k + m_{k-1} p^{k-1} + \cdots + m_1 p + m_0$ and $n = n_k p^k + n_{k-1} p^{k-1} + \cdots + n_1 p + n_0$.

Before proving the Lucas' Theorem, we first prove some basic facts.

Theorem 4.1 $\binom{p}{n} \equiv 0 \pmod{p}$ with $1 \leq n \leq p-1$
Proof

$$\binom{p}{n} = \frac{p(p-1)(p-2)\dots(p-n+1)}{n(n-1)\dots 1} \Rightarrow p \mid \binom{p}{n}$$

Theorem 4.2 $(1+X)^p = 1 + X^p \pmod{p}$
Proof

$$\begin{aligned} (1+X)^p &= 1 + \tbinom{p}{1}X + \tbinom{p}{2}X^2 + \cdots + \tbinom{p}{p-1}X^{p-1} + X^p && \pmod{p} \\ &= 1 + X^p && \pmod{p} \\ &\quad \text{As all other binomial coefficient equal to 0} \end{aligned}$$

Theorem 4.3 $(1+X)^{p^k} = 1 + X^{p^k} \pmod{p}$
This can be proved by Mathematical Induction.
We have proved the case when $k = 1$.
Assume the statement is true for some positive integer k.

$$\begin{aligned} (1+X)^{p^{k+1}} &= \left[(1+X)^{p^k}\right]^p && \pmod{p} \\ &= (1+X^{p^k})^p && \pmod{p} \\ &= 1 + X^{p^{k+1}} && \pmod{p} \end{aligned}$$

Proof of Lucas' Theorem This proof based on generating function
Consider

$$\begin{aligned} \sum_{n=0}^{m} \binom{m}{n} X^n &= (1+X)^m \\ &= \prod_{i=0}^{k} \left[(1+X)^{p^i}\right]^{m_i} && \text{As } m = \textstyle\sum p^i m_i \\ &= \prod_{i=0}^{k} (1+X^{p^i})^{m_i} && \text{Theorem 4.3} \\ &= \prod_{i=0}^{k} \sum_{n_i=0}^{m_i} \binom{m_i}{n_i} X^{p^i n_i} && \text{Binomial theorem} \\ &\quad \text{expand the terms, we obtain the following} \\ &= \left[\tbinom{m_0}{0}X^{0p^0} + \tbinom{m_0}{1}X^{1p^0} + \cdots + \tbinom{m_0}{m_0}X^{m_0p^0}\right] \times \\ &\quad \left[\tbinom{m_1}{0}X^{0p^1} + \tbinom{m_1}{1}X^{1p^1} + \cdots + \tbinom{m_1}{m_1}X^{m_1p^1}\right] \times \\ &\quad \cdots \times \\ &\quad \left[\tbinom{m_k}{0}X^{0p^k} + \tbinom{m_k}{1}X^{1p^k} + \cdots + \tbinom{m_k}{m_k}X^{m_kp^k}\right] \\ &\quad \text{consider the coefficient of } X^n \\ &= \sum_{n=0}^{m} \left[\prod_{i=0}^{k} \binom{m_i}{n_i}\right] X^n \end{aligned}$$

5 Möbius Inversion Function

Definiton The definition of Möbius Function as follow

$$\mu(n) = \begin{cases} (-1)^k & \text{if } n \text{ is square-free integer, } n = p_1 p_2 \dots p_k \\ 0 & \text{o.t.w.} \end{cases}$$

Note that by definition, $\mu(1) = 1$. Also, Möbius Function is a multiplicative function.

Problem Chiu Chiu Star Problem

Starry Starry night, Chiu Chiu is sitting at the origin of a 3-dimensional cartesian coordinate system. On every integer point (x, y, z) in $[1, a] \times [1, b] \times [1, c]$, there is a lovely star.You can imagine that some of the stars are blocked by others sadly.
As a star lover, Chiu Chiu want to know how many star he can see directly.

$Solution: \sum_{i=0}^{min(a,b,c)} \mu(i) \lfloor \frac{a}{i} \rfloor \lfloor \frac{b}{i} \rfloor \lfloor \frac{c}{i} \rfloor$
For a given star (x, y, z), it can be seen if and only if $gcd(x, y, z) = 1$. Otherwise, it will be blocked by star $(\frac{x}{gcd(x,y,z)}, \frac{y}{gcd(x,y,z)}, \frac{z}{gcd(x,y,z)})$.
The problem statement becomes 'count the number of $1 \le x \le a$, $1 \le y \le b$ and $1 \le z \le c$ such that $gcd(x, y, z) = 1$'.

Theorem 5.1 $\lfloor \frac{a}{i} \rfloor \lfloor \frac{b}{i} \rfloor \lfloor \frac{c}{i} \rfloor = \#$ of $1 \le x \le a$, $1 \le y \le b$ and $1 \le z \le c$ such that $i | gcd(x, y, z)$ *Proof*
First, we know that $i | gcd(x, y, z) \Leftrightarrow i | x$ and $i | y$ and $i | z$.
There are $\lfloor \frac{a}{i} \rfloor$ numbers within $[1, a]$ which is divisible by i. They are $i, 2i, \dots, i\lfloor \frac{a}{i} \rfloor$.
Similar for b and c.
So, there are $\lfloor \frac{a}{i} \rfloor \lfloor \frac{b}{i} \rfloor \lfloor \frac{c}{i} \rfloor$ of such points.

Denote $F\{p_1, p_2, \dots, p_k\} = \#$ of points $1 p_1 p_2 \dots p_k \mid gcd(x, y, z)$.
By inclusion-exclusion primciple, we obtain that number of points with $p \nmid gcd(x, y, z)$ for all prime p (or $gcd(x, y, z) = 1$) is equal to:
# of points $1 \mid gcd(x, y, z)$ - # of points $p_1 \mid gcd(x, y, z)$ + # of points $p_1 p_2 \mid gcd(x, y, z) + \dots$

$$\begin{aligned} \text{Answer} &= F\{\phi\} - \sum_{p_1} F\{p_1\} + \sum_{p_1, p_2} F\{p_1, p_2\} - \sum_{p_1, p_2, p_3} F\{p_1, p_2, p_3\} + \dots + (-1)^k F\{p_1, p_2, \dots, p_k\} \\ &= \sum_{i=1}^{min(a,b,c)} \begin{cases} (-1)^k F(i) & \text{if } i \text{ is square-free integer, } i = 1 p_1 p_2 \dots p_k \\ 0 & \text{o.t.w.} \end{cases} \\ &\quad \text{As each number } i = p_1 p_2 \dots p_k \text{ corresponds to a square free integer within } min(a, b, c) \\ &= \sum_{i=1}^{min(a,b,c)} \mu(i) \lfloor \frac{a}{i} \rfloor \lfloor \frac{b}{i} \rfloor \lfloor \frac{c}{i} \rfloor \end{aligned}$$

Before moving on to more properties of Möbius Function. Here, we introduce a new function.

$$I(n) = \sum_{d | n} \mu(d)$$

Prop. $I(n)$ is a multiplicative function
Proof We here prove a stronger statement.

Theorem 5.2 Let

$$F(n) = \sum_{d|n} f(d)$$

then f is multiplicative $\Rightarrow F$ is multiplicative. Consider two positive integers n, m which co-prime to each other.

$$\begin{aligned} F(n)F(m) &= \left[\sum_{c|n} f(c)\right]\left[\sum_{d|m} f(d)\right] \\ &= \sum_{c|n}\sum_{d|m} f(c)f(d) \\ &= \sum_{cd|nm} f(cd) \\ &= F(nm) \end{aligned}$$

As $\mu(n)$ is a multiplicative function $I(n)$ also multiplicative.

Theorem 5.3 $I(n) = \begin{cases} 1 & \text{if } n = 1 \\ 0 & \text{if } n \neq 1 \end{cases}$

Proof As $I(n)$ is multiplicative, we only need to show when $n = 1$ and $n = p^k$ where p is a prime and k is a positive integer.
When $n = 1$, $I(1) = \mu(1) = 1$
When $n = p^k$, $I(n) = \mu(1) + \mu(p) + \mu(p^2) + \cdots + \mu(p^k) = 1 - 1 + 0 + \cdots + 0 = 0$

We will use this result later.

Theorem 5.4 Möbius Inversion Formula: $F(n) = \sum_{d|n} f(d) \Leftrightarrow f(n) = \sum_{d|n} \mu(d)F(\frac{n}{d})$

The result can also be written as $f(n) = (\mu * F)(n)$, the Dirichlet convolution μ and F.
Proof $\Rightarrow$

$$\begin{aligned} & \sum_{d|n} \mu(d)F(\frac{n}{d}) & \\ = & \sum_{d|n} \mu(d) \sum_{c|\frac{n}{d}} f(c) & \\ = & \sum_{d|n}\sum_{c|\frac{n}{d}} \mu(d)f(c) & \\ = & \sum_{c|n}\sum_{d|\frac{n}{c}} \mu(d)f(c) & \\ = & \sum_{c|n} f(c) \sum_{d|\frac{n}{c}} \mu(d) & \text{Theorem 5.3} \\ = & \sum_{c|n} f(c)I(\frac{n}{c}) & \star \text{ this term becomes 1 if and only if } c = n \\ = & f(n) & \end{aligned}$$

Proof of another direction leave to the readers.

Application Last week we learnt how to apply Burnside Lemma into a necklace problem. We now have a sub-problem.
Assume there is a necklace with n jewelries, there are r kinds of jewelry each have infinitely many of quantity. How many combination of them are identical under rotating with period d? Let it as

$f(d)$. We know that $\sum_{d|n} d\, f(d) = r^n$

By Möbius Inversion,

$$f(n) = \frac{1}{n} \sum_{d|n} \mu(d) r^{\frac{n}{d}}$$

Theorem 5.5 One of the generalization of Möbius Inversion Fomula

$$g(x) = \sum_{i=1}^{\infty} f(ix) \Leftrightarrow f(x) = \sum_{i=1}^{\infty} \mu(i) g(ix)$$

Assume that g and f are absolutely convergent.

Proof $\Rightarrow$

$$\begin{array}{rl|l} & \sum_{i=1}^{\infty} \mu(i) g(ix) & \\ = & \sum_{i=1}^{\infty} \mu(i) \sum_{j=1}^{\infty} f(ijx) & \\ = & \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \mu(i) f(ijx) & \\ = & \sum_{u=1}^{\infty} f(ux) \sum_{v|u} \mu(v) & \text{Change of variable, } u = ij \\ = & \sum_{u=1}^{\infty} f(ux) I(u) & \text{Theorem 5.3} \\ = & f(x) & \end{array}$$

Proof of another direction leave to the readers.

We now look back to Chiu Chiu Star Problem.
Let $g(n) := \#\{i : i \mid gcd(x, y, z)\} = \lfloor \frac{a}{i} \rfloor \lfloor \frac{b}{i} \rfloor \lfloor \frac{c}{i} \rfloor$
Let $f(n) := \#\{i : gcd(x, y, z) = i\}$
Clearly, $g(n) = \sum_{i=1}^{\infty} f(in)$. Therefore, by Theorem 5.5,

$$f(1) = \sum_{i=1}^{\infty} \mu(i) g(i)$$

6 Euler Phi Function

Definition $\varphi(n) := \#\{m : (n, m) = 1 \text{ and } m \in [0, n)\}$
Here is the result of phi function of first few integers.

x	1	2	3	4	5	6	7	8	...
$\varphi(x)$	1	1	2	2	4	2	6	4	...

Here, we investigate some basic properties of Euler Phi Function.

Theorem 6.1 For a prime P, $\varphi(P) = P - 1$

Proof Obviously, for a prime number P, all of integers from 1 to $P - 1$ co-prime to itself.

Theorem 6.2 For a prime P, $\varphi(P^k) = P^k - P^{k-1}$

Proof Consider power of a prime P^k. All of the integers q which is **not** co-prime to P^k must have a factor of P, or must be divisible by P. There are $P^k/P = P^{k-1}$ of such numbers.

Theorem 6.3 Euler Phi Function is multiplicative
i.e. $\varphi(nm) = \varphi(n)\varphi(m)$ for all n, m co-prime to each other
Proof Consider integers from 0 to $nm - 1$

$$\begin{array}{cccc} 0 & 1 & \dots & m-1 \\ m & m+1 & \dots & 2m-1 \\ \vdots & \vdots & \ddots & \vdots \\ (n-1)m & (n-1)m+1 & \dots & nm-1 \end{array}$$

Here is an important fact. For an integer x co-prime to nm, it must be both co-prime to n and m. Proof as follow.

$$\left.\begin{array}{rrcl} & (x, nm) & = & 1 \\ \Leftrightarrow & ix + jnm & = & 1 \\ \Leftrightarrow & ix + (jm)n = 1 & \& & ix + (jn)m = 1 \\ \Leftrightarrow & (x, n) = 1 & \& & (x, m) = 1 \end{array}\right| \text{Extended Euclidean Algorithm}$$

First, we count the number of columns that co-prime to m. This action based on a fact that if one element in that column co-prime to m, all other elements also co-prime to m.
i.e. $(km + r, m) = 1 \Leftrightarrow (r, m) = 1$

$$\begin{array}{rrcl} & (km + r, m) & = & 1 \\ \Leftrightarrow & i(km + r) + jm & = & 1 \\ \Leftrightarrow & ir + (j + k)m & = & 1 \\ \Leftrightarrow & (r, m) & = & 1 \end{array}$$

The number of columns co-prime to m equals to the number of integers co-prime to m between 1 and m. Which is $\varphi(m)$.

As we know the number of columns co-prime to m, we want to know the number of elements co-prime to n in each column. Consider a column, all numbers in the column can be written as $km + r$ for a fixed r and $0 \le k \le n - 1$ uniquely.
Here, we prove that $(x, n) = 1 \Leftrightarrow (x \mod n, n) = 1$. The proof is similar to the above.

$$\begin{array}{rrcl} & (x \mod n, m) & = & 1 \\ \Leftrightarrow & i(x - kn) + jn & = & 1 \\ \Leftrightarrow & ix + (j - ik)n & = & 1 \\ \Leftrightarrow & (x, n) & = & 1 \end{array}$$

As $(n, m) = 1$, $km + r = lm + r \pmod n \Rightarrow km = lm \pmod n \Rightarrow k = l \pmod n$. Therefore no two element in the same column are the same under modulo n. i.e.

$$\{r, m + r, 2m + r, \dots, (n - 1)m + r\} \equiv \{1, 2, \dots, n\} \pmod n$$

and the number of elements co-prime to n is $\varphi(n)$.

As there are $\varphi(n)$ elements in $\varphi(m)$ columns co-prime to nm, $\varphi(nm) = \varphi(n)\varphi(m)$.

To compute the Euler Phi Function quickly, we can make use of the following property.

Theorem 6.4 $n = \sum_{d|n} \varphi(d)$

Let $F(n) = \sum_{d|n} \varphi(d)$. Note that by theorem 5.2, $F(n)$ is a multiplicative function since $\varphi(n)$ is multiplicative, we only need to consider case $n = p^k$ with prime k and positive integer index k.
$F(p^k) = \varphi(1) + \varphi(p) + \varphi(p^2) + \cdots + \varphi(p^k) = p^0 + (p^1 - p^0) + (p^2 - p^1) + \cdots + (p^k - p^{k-1}) = p^k = n$

Rewritting the above equation into $\varphi(n) = n - \sum_{d|n \text{ and } d \neq n} \varphi(d)$. Value of Euler Phi Function from 1 to n can be compute in $\Theta(n \log n)$ time.

Psuedocode Computing Euler Phi Function

1: **for all** elements in *phi* **do**
2: $phi[i] = i$
3: **end for**
4: **for** $i = 1, 2, \ldots, n$ **do**
5: **for** $j = 2i, 3i, \ldots, \lfloor \frac{n}{i} \rfloor i$ **do**
6: $phi[j] = phi[j] - phi[i]$
7: **end for**
8: **end for**

Or C++ code as below.

```
int phi[MAX];
for (int i = 1; i < MAX; i++) phi[i] = i;
for (int i = 1; i < MAX; i++){
        for (int j = i << 1; j < MAX; j += i){
                phi[j] -= phi[i];
        }
}
```

Fermat Little Theorem $a^{P-1} = 1 \pmod P$ for all $1 \leq a \leq P-1$ and prime P

Fermat-Euler Theorem This give a more general statement

$$a^{\varphi(n)} = 1 \pmod n$$

if a and n co-prime. We call such a as a root.
Actually, Fermat Little Theorem is a special case of it.
Proof Here is a direct proof copied from Wikipedia
Let $R = x_1, x_2, \ldots, x_{\varphi(n)}$ be a reduced residue system $\pmod n$ and a as an integer co-prime to P.

$$ax_i \equiv ax_j \pmod n \Rightarrow x_i \equiv x_j \pmod n \Rightarrow i = j$$

Therefore, $R \equiv aR \pmod{n}$.
By multiplying all elements in R and aR, we have the following result.

$$\begin{aligned} \prod_{i=0}^{\varphi(n)} x_i &\equiv \prod_{i=0}^{\varphi(n)} ax_i && \pmod{n} \\ 1 \times \prod_{i=0}^{\varphi(n)} x_i &\equiv a^{\varphi(n)} \times \prod_{i=0}^{\varphi(n)} x_i && \pmod{n} \\ 1 &\equiv a^{\varphi(n)} && \pmod{n} \end{aligned}$$

The meaning of this theorem is that, in a residue system modulo n, power of a number a co-prime to n have a cycle of $\varphi(n)$ (Notice that it may not be the least cycle).

Accroding to Fermat Little Theorem,

$$a^{P-1} \equiv 1 \pmod{P}$$

is true for all prime P and $1 \leq a \leq P-1$.
For example, when $P = 7$ and $n = 2$

$$\begin{aligned} &2\,(2 \mod 7) \to 4\,(4 \mod 7) \to 1\,(8 \mod 7) \to \\ &2\,(16 \mod 7) \to 4\,(32 \mod 7) \to 1\,(64 \mod 7) \end{aligned}$$

We observe that $2^k \mod 7$ form a cycle. The length, by Fermat Little Theorem, is $P - 1 = 6$. However, this may not be the least cycle.

By the above observation, we have the following theorem.

Theorem 6.5 If a and n co-prime, we have the following

$$a^k \equiv a^{k \pmod{\varphi(n)}} \pmod{n}$$

Proof Here we first give a proof of a weaker statement

$$a^k \equiv a^{k-\varphi(n)} \pmod{n}$$

Since $a^{\varphi}(n) \equiv 1 \pmod{n}$ accroding to Fermat-Euler Theorem.

$$a^{k-\varphi(n)} \equiv a^k \div a^{\varphi(n)} \equiv a^k \pmod{n}$$

By induction, the above theorem is true.

Problem Uva 10692 Huge Mod
Given a sequence of integer $(a_1, a_2, a_3, \ldots, a_n)$ and m, compute

$$a_1^{a_2^{\cdot^{\cdot^{a_n}}}} \mod m$$

7 Primitive Root

道生一，一生二，二生三，三生萬物。 《老子》

In this section, we discuss the properties of power of integer under modulo a number n, usually a prime.

An integer a under modulo n is called a Root of Unity if it's least cycle is exactly $\varphi(n)$. i.e. $(a, a^2, a^3, \ldots, a^{\varphi(n)})$ are distinct and $a^{\varphi(n)} \equiv 1 \pmod{n}$.
When n is a prime, denote as P, we know that $\varphi(P) = P - 1$. Such number a is called a Primitive Root. It will generate all numbers between 1 and $P - 1$ inclusively.
i.e. if a is a primitive root

$$\{a^1, a^2, \ldots, a^{p-1}\} \equiv \{1, 2, \ldots, P - 1\} \pmod{P}$$

As there are no quick method for computing a root of unity(so far), the method we usually use is randomly guess a number :o). We will introduce some methods to check weather a number is a root of unity. Also with analysis of the algorithm.

After guessing a number, we need to check weather $a^k = 1 \pmod{n}$ for any $0 < k < \varphi(n)$. If such k exsist, it is not a root of unity. However, checking all k will cost $\Theta(n \log n)$. The following theorem provide a method to speed up the checking process.

Theorem 7.1 if a is a number co-prime to n

$$a^k \equiv 1 \pmod{n} \Rightarrow k \mid \varphi(n)$$

with least k. *Proof* Here we prove a stronger statement $a^h \equiv 1 \pmod{n} \Rightarrow k \mid h$.
We proof this by contradiction. Assume that $k \nmid h$ or $h = qk + r$. with $0 < r < k$.

$$1 = a^h = a^{qk+r} = (a^k)^q(a^r) = a^r$$

Since k is the least positive integer such that $a^k \equiv 1 \pmod{n}$, a^r must not equal to 1. This lead to a contradiction.

We have just proved that $a^h \equiv 1 \Rightarrow h \mid k$. As $\varphi(n)$ is one of the h satisfying the above condition, $k \mid \varphi(n)$ holds.

This theorem give us a hint. If a is not a root of unity, i.e. $a^k = 1 \pmod{n}$ for some $k < \varphi(n)$, such k must be a divisor of $\varphi(n)$. Therefore, checking a^k for all k as the divisor of $\varphi(n)$ is sufficiently enough. But we are still not satisfied.

Theorem 7.2 if there exist a k such that $a^k = 1 \pmod{n}$ and $0 < k < \varphi(n)$

$$a^{\frac{\varphi(n)}{p}} = 1 \pmod{n}$$

for some p as a prime factor of $\varphi(n)$. *Proof* By theorem 7.1, we know that $k'k = \varphi(n)$ for some integer $k' > 1$. Write k' as pq where p is a prime factor of k', which also be a prime factor of $\varphi(n)$.Then

$$a^{\frac{\varphi(n)}{p}} = a^{k\frac{k'}{p}} = (a^k)^q = 1 \pmod{n}$$

By this theorem, we can check weather a number is a root of unity just by checking $a^{\frac{\varphi n}{p}}$ for all prime factors of $\varphi(n)$.

Theorem 7.3 A primitive root a generate a equivalent class of all numbers co-prime to n under modulo n. i.e.

$$\{a, a^2, a^3, \ldots, a^{\varphi(n)}\} \equiv \{d : (d, n) = 1\} \pmod{n}$$

Proof By Fermat-Euler Theorem, numbers on the right are all of the roots.
Any number on the left a^k, as $a^{k\varphi(n)} \equiv 1 \pmod{n}$, are also roots.
Therefore, two classes equivalent.

Now, here is a question. "What is the probability that I successfully guessed a root of unity?". To answer the question, we need to know the number of root of unity.

Theorem 7.4 number of root of unity under modulo n is $\varphi(\varphi(n))$
Proof By theorem 7.3, we know that all roots can be written in a form $a^k \pmod{n}$ as a is a primitive root, including primitive root.
If a^k is also a primitive root

$$\{a^k, a^{2k}, \ldots, a^{\varphi(n)k}\} \equiv \{d : (d, n) = 1\} \pmod{n}$$

As it goes with a cycle of length $\varphi(n)$ by theorem 6.5,

$$\{k, 2k, \ldots, \varphi(n)k\} \equiv \{1, 2, \ldots, \varphi(n)\} \pmod{\varphi(n)}$$

i.e. $ik = jk \pmod{\varphi(n)} \Leftrightarrow i = j$, for all $1 \leq i, j \leq \varphi(n)$
We obtain that $(k, \varphi(n)) = 1$ in order to have this property. The number of k co-prime to $\varphi(n)$ is $\varphi(\varphi(n))$.

Therefore, the number of root of unity is exactly $\varphi(\varphi(n))$.

Note that this result is based on an assumption that there exist at least one root of unity. For prime, this is true, there always exist a primitive root. You can search the proof on the internet.

Problem POJ 1284 Primitive Roots

8 References

Content of this note is copied from everywhere on the internet. Special thanks to Google Search and Wikipedia.

I have read (and copied) some lecture notes from J. Owen Sizemore, please visit his course webpage if you are interested. Especially the last few lectures.
http://www.math.wisc.edu/ josizemore/Numbertheory567.html

⋆ This note is only for CUHK ACM Team Training, not an academic paper.⋆