Number Theory

Fermat's Little theorem

Fermat’s Little Theorem

- Fermat's little theorem states that if p is a prime number, then for any integer a, the number $a^p - a$ is an integer multiple of p.

$$a^p \equiv a \pmod{p}$$

Fermat’s Little Theorem

- If a is not divisible by p, Fermat's little theorem is equivalent to the statement that $a^{p-1} - 1$ is an integer multiple of p

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

Modulo inverse using Fermat's Theorem

$$a^p \equiv a \pmod{p}$$

If a is not divisible by p,

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

Multiplying both sides by a^{-1}

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

Evaluating $(a/b)\%m = (a\%m) * (b^{-1}\%m)$

- $(a/b) \% m = ((a \% m) * (b^{-1} \% m)) \% m$

Euler Totient Function

Euler Totient Function

Represented as $\Phi(n)$ → $1 \leq m < n$, $\gcd(m, n) = 1$

↑ number of numbers from 1 to n-1 which are coprime with n

$$\Phi(n) = n*(1 - 1/p_1)*(1 - 1/p_2)*(1 - 1/p_3)...*(1 - 1/p_k)$$

where $p_1, p_2, p_3, .. , p_k$ are distinct prime factors of n.

Euler Totient Function - Derivation

If A and B are coprime or gcd(A,B) = 1 then, $\Phi(A*B) = \Phi(A)*\Phi(B)$

We can write

$$n = p_1^a * p_2^b * p_3^c \ldots p_k^k$$
$$\Phi(n) = \Phi(p_1^a * p_2^b * p_3^c \ldots p_k^k)$$

Since $\gcd(p_1^a, p_2^b) = 1$

$$\Phi(n) = \Phi(p_1^a) * \Phi(p_2^b) * \Phi(p_3^c) * \ldots * \Phi(p_k^k)$$

Euler Totient Function - Derivation

Let us analyze $\Phi(p^a)$

Numbers from 1 to p^a which are not coprime with p^a are p, 2p, 3p... p^a.

This is an AP with common difference p and first term p. Using the formula of nth term of AP, we get

$$p^a = p + (x-1)*p$$

$$x = p^{a-1}$$

Therefore, the number of numbers that are coprime with p^a are

$$p^a - p^{a-1}$$

$$p^a (1 - 1/p)$$

Euler Totient Function - Derivation

Substituting this in above equation of $\Phi(n)$, we get

$$\Phi(n) = p_1^{a}(1-1/p_1) * p_2^{b}(1-1/p_2) * p_3^{c}(1-1/p_3) * \ldots * p_k^{k}(1-1/p_k)$$

$$\Phi(n) = p_1^{a} * p_2^{b} * p_3^{c} * p_k^{k} * (1-1/p_1) *(1-1/p_2) *(1-1/p_3) * \ldots *(1-1/p_k)$$

$$\Phi(n) = n * (1-1/p_1) *(1-1/p_2) *(1-1/p_3) * \ldots *(1-1/p_k)$$

since $n = p_1^{a} * p_2^{b} * p_3^{c} *\ldots* p_k^{k}$

Euler's Totient Theorem

If a and n have no common divisors, then

$$a^{\phi(n)} \equiv 1 \pmod{n}$$

Euler's Totient Theorem

If a and n have no common divisors, then

$$a^{\phi(n)} \equiv 1 \pmod{n}$$

$$a^b \mod m = a^{b \mod \phi(m)} \mod m$$

Extended Euclidean Algorithm

Diophantine Equations

These are the polynomial equations for which integral solution exists

Example: $3x + 7y = 1$ or $x^2 - y^2 = z^3$

$$ax + by = c$$

Note: $a,b,c \in I$

Solutions to these equations exist only if gcd(a,b) divides c.

Extended Euclid Algorithm

GCD(a,b) has the property that

$$ax + by = gcd(a,b)$$

Now, we have to find the values of x and y..

Extended Euclid Algo

$ax + by = gcd(a, b)$

$gcd(a, b) = gcd(b, a\%b)$

$gcd(b, a\%b) = bx_1 + (a\%b)y_1$

$a\%b = a - (a/b) * b$

From the above equations we get,

$ax + by = bx_1 + (a\%b)y_1$

$ax + by = bx_1 + (a - (a/b) * b)y_1$

$ax + by = ay_1 + b(x_1 - (a/b) * y_1)$

Extended Euclid Algo

$ax + by = gcd(a, b)$
$gcd(a, b) = gcd(b, a\%b)$
$gcd(b, a\%b) = bx_1 + (a\%b)y_1$
$a\%b = a - (a/b) * b$

From the above equations we get,

$ax + by = bx_1 + (a\%b)y_1$
$ax + by = bx_1 + (a - (a/b) * b)y_1$
$ax + by = ay_1 + b(x_1 - (a/b) * y_1)$

Comparing the coefficients of a and b, we get

$$x = y_1$$
$$y = x_1 - (a/b) * y_1$$

```cpp
ll extended_euclid(ll a, ll b, ll &x, ll &y) {
  if (b == 0) {
    x = 1; y = 0;
    return a;
  }
  ll x1, y1;
  ll d = extended_euclid(b, a % b, x1, y1);
  x = y1;
  y = x1 - y1 * (a / b);
  return d;
}

ll inverse(ll a, ll m) {
  ll x, y;
  ll g = extended_euclid(a, m, x, y);
  if (g != 1) return -1;
  return (x % m + m) % m;
}
```