Number theory

Leonardo Cervantes

May, 2023

1 Prime numbers

Definition 1.1. *A number p is prime if the only positive divisors of p are 1 and p.*

Definition 1.2. *A number n is composite if n is not prime.*

Test if a number is prime

Check if n is divisible by any number a such that $a \leq \sqrt{n}$.

GCD

The greatest common divisor of two integers a and b is the largest integer d that divides both a and b.
The GCD of a and b is denoted by $\gcd(a, b)$ and can be computed as follows:

$$\gcd(a,b) = \begin{cases} a & \text{if } b = 0 \\ \gcd(b, a \mod b) & \text{if } b \neq 0 \end{cases}$$

A way to implement this in C++ is the following:

```
int gcd(int a, int b) {
    if (b == 0) return a;
    return gcd(b, a % b);
}
```

LCM

The least common multiple of two integers a and b is the smallest positive integer that is divisible by both a and b.
The LCM of a and b is denoted by $\text{lcm}(a, b)$ and can be computed as follows:

$$\text{lcm}(a,b) = \frac{a \cdot b}{\gcd(a,b)}$$

2 Modular arithmetic

Definition 2.1. *Let a and b be integers and n a positive integer. We say that a is congruent to b modulo n if n divides $a - b$. We write $a \equiv b \mod n$.*

Inverse modulo k

$$a \cdot a^{-1} \equiv 1 \mod k$$

Fermat's little theorem

If p is a prime number, then for any integer a, the number $a^p - a$ is an integer multiple of p.

$$a^p \equiv a \mod p$$

Moreover, if a is not divisible by p, then

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

Then, we can deduce by this that:

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

This is important for competitive programming beecause it allows us to compute the inverse of a number modulo p in $O(\log p)$. However, this is not the fastest way to compute the inverse of a number modulo p. Another way is to use the extended Euclidean algorithm.

Extended Euclidean algorithm

The extended Euclidean algorithm is an efficient way to find integers x and y such that

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

Then if $\gcd(a, b) = 1$, we can find the inverse of a modulo b by finding x and y such that

$$ax + by = 1$$

To find x and y, we can use the following function:

```
long long inverse(long long a,long long b,long long n,long long m){
    if(a==1){
        return n;
    }
    if(a<b){
        long long x=b/a;
        m+=(xn);
        m=m%MOD;
        b=b%a;
        return inverse(a,b,n,m);
    }
```

```
        else if(b==1){
            return(MOD-m);
        }
        else{
            long long x=a/b;
            n+=(xm);
            n=n%MOD;
            a=a%b;
            return inverse(a,b,n,m);
        }
    }
```