Introducción a teoría de números

PCFIM

Modular exponentiation: $O(\log y)$

$$x^y \mod p$$

```
ll fastexp(ll x, ll y, ll p){
        ll ans = 1;
        while(y > 0){
                if(y&1) ans = (ans*x)%p;
                y = y>>1;
                x = (x*x)%p;
        }
        return ans%p;
}
```

Eratosthenes Sieve: $O(N \log \log N)$

```
bool isPrime[N];
void sieve(){
        for(int i=2; i<N; i++){
                if(!isPrime[i]){
                        for(ll j=1LL*i*i; j<N; j+=i) isPrime[j] = 1;
                }
        }
}
```

Modified sieve

Least Prime Factor: $O(N \log \log N)$

$$\text{lpf}[i] : \text{Menor factor primo que divide a } i$$

```
int lpf[N];
void sieve(){
        for(int i=2; i<N; i++){
                if(!lpf[i]){
                        lpf[i] = i;
                        for(ll j=1LL*i*i; j<N; j+=i){
                                if(lpf[j] == 0) lpf[j] = i;
                        }
                }
        }
}
```

Euler's Totient Function: $O(N \log \log N)$

$\varphi(n)$ es la cantidad de numeros coprimos con n en el rango de 1 a n.

$$\varphi(n) = n \prod_{p|n} \left(1 - \frac{1}{p}\right)$$

```
int phi[N];
void cphi(){
        phi[1] = 1;
        for(int i=2; i<N; i++){
                if(!phi[i]){
                        phi[i] = i-1;
                        for(ll j=2*i; j<N; j+=i){
                                if(phi[j] == 0) phi[j] = j;
                                phi[j] = phi[j]/i*(i-1);
                        }
                }
        }
}
```