Maths for Competitive Programming

Content

Modular Arithmetic

Why is it Used?

Well holding down values consume memory and maximum integer size available in c++ is of order 10^{18}

But what if calculations are going over 10^{18}, then we store the value in modular form

Formulaes

1. $(a * b)\%m = ((a\%m) * (b\%m))\%m$
2. $(a + b)\%m = ((a\%m) + (b\%m))\%m$

Sample

1. $(32 * 25)\%3$
   Normal method = $800\%3$ = 2
   Modular Arithmetic = $((32\%3) * (25\%3))\%3$ = $(2 * 1)\%3$ = 2

Modular Binary Exponentiation

Let's say we need to find $(a^b)\%m$

One simple way is to

```
long long int ans = 1
for(int i = 1; i <= b; ++i){
    ans = ((ans%m)*(a%m))%m;
    //As we learned in Modular Arithmetic
}
```

But this method is $O(n)$, this method might be good but we need a better method

Let's remember mathematics a little

$a^n = (a^2)^{n/2}$ ... if n is even

$a^n = a.(a^2)^{(n-1)/2}$ ... if n is odd

Therefore we can find a^n in $O(log_2(n))$

```
long long int binary_modular_exponentiation(long long int a, long long int n){
    if(n == 0)return 1;
    if(n == 1)return a;
    long long int k = binary_modular_exponentiation(a*a,n/2);
    if(n%2 == 1){
        k*=a;
    }
    return k;
}
```