数论

gcd_lcm

```
int gcd(int a,int b){
   return (b ? gcd(b,a%b) : a);
}

int lcm(int a,int b){
   return a/gcd(a,b)*b;
}
```

组合数

n, m 较小

用二维数组的杨辉三角模拟
$C[i][j] = C_i^j$

```
c[0][0] = 1;
for(int i=0;i<=k;++i)
{
    for(int j=0;j<=i;++j)       C[i][j] = (C[i-1][j-1] + C[i-1][j])%M;
}
```

n,m 较大

$$C_n^m = \frac{n!}{(n-m)!\cdot m!}$$

逆元分数即可。

```
const int N = 2e5+10;
const int M = 1e9+7;

times[0] = 1;// 当 n = m 时，t[n-m] 应该为 1 而不是 0。
for(int i=1;i<=N;++i)   times[i] = (times[i-1]*i)%M;

int times[N];
int qpow(int a,int b) // 快速幂
{
        int z = 1;
        while(b)
        {
                if(b&1) z = (z*a)%M;
                a = (a*a)%M;
                b >>= 1;
        }
        return z;
}

int inv(int x) // 逆元
{
        return qpow(x,M-2);
}

int C(int x,int y) // 组合数
{
        return (times[x]*inv(times[x-y]*times[y]%M)%M);
}
```