

数论 5

卢卡斯定理

算法简介

$O(p + \log_p n)$ 计算 $\binom{n}{m} \bmod p$ 的算法。

算法实现

$$\binom{p}{i} = \frac{p!}{i!(p-i)!}$$

于是对 $1 \leq i \leq p-1$ 有 $\binom{p}{i} \equiv 0 \pmod p$

$$(1+x)^p \equiv \binom{p}{0} + \binom{p}{1}x + \dots + \binom{p}{p}x^p \equiv 1+x^p \pmod p$$

对 $\binom{n}{m}$ 不妨设 $n=k_1p+r_1, m=k_2p+r_2$ 于是有

$$(1+x)^n \equiv (1+x)^{k_1p+r_1} \equiv ((1+x)^p)^{k_1} (1+x)^{r_1} \equiv (1+x^p)^{k_1} (1+x)^{r_1} \pmod p$$

对比左右两边 x_m 的系数，有

$$\binom{n}{m} \equiv \binom{k_1}{k_2} \binom{r_1}{r_2} \pmod p$$

于是递归处理，有

$$\text{Lucas}(n, m) \equiv \text{Lucas}\left(\left\lfloor \frac{n}{p} \right\rfloor, \left\lfloor \frac{m}{p} \right\rfloor\right) \binom{n \bmod p}{m \bmod p}$$

代码模板

[洛谷p3807](#)

```
const int MAXN=1e5+5;
int frac[MAXN],invfrac[MAXN];
int quick_pow(int a,int b,int p){
    int ans=1;
    while(b){
        if(b&1)ans=1LL*ans*a%p;
        a=1LL*a*a%p;
        b>>=1;
    }
    return ans;
}
int Lucas(int n,int m,int p){
```

```
if(n<m)return 0;
else if(n<p)return 1LL*frac[n]*invfrac[n-m]%p*invfrac[m]%p;
else return 1LL*Lucas(n%p,m%p,p)*Lucas(n/p,m/p,p)%p;
}
int main()
{
    int T=read_int();
    while(T--){
        int n=read_int(),m=read_int(),p=read_int();
        frac[0]=1;
        _for(i,1,p)frac[i]=1LL*frac[i-1]*i%p;
        invfrac[p-1]=quick_pow(frac[p-1],p-2,p);
        for(int i=p-1;i;i--)
            invfrac[i-1]=1LL*invfrac[i]*i%p;
        enter(Lucas(n+m,n,p));
    }
    return 0;
}
```

拓展卢卡斯

From: <https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link: https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:jxm2001:%E6%95%B0%E8%AE%BA_5&rev=1602814810

Last update: 2020/10/16 10:20