P3807 【模板】卢卡斯定理

传送门

卢卡斯定理$(Lucas)$：用来求解组合数取模问题：$C(n,m)\%p$

这里直接用百度的定义和推导：~~(我觉得讲的很清晰了)~~

定律定义

Lucas定理：我们令***n=sp+q , m=tp+r***．（***q*** ，***r ≤p***）

那么：

$$\binom{sp+q}{tp+r} \equiv \binom{s}{t}\binom{q}{r}\ (mod\ (p$$

（在编程时你只要继续对 $\binom{s}{t}$ 调用Lucas定理即可。

代码可以递归的去完成这个过程，其中递归终点为***t = 0***；

时间***O***（***$log_p(n)*p$)***：）

推导过程:

Lucas定理证明：

首先你需要这个算式: $\binom{p}{f} \equiv 0\,(mod\,(p))$,其中f > 0&& f < p，然后

(1 + x) ^{n} **Ξ*(1 + x)*** $^{sp+q}$ **Ξ*((1 + x)*** ^{p} ***)*** ^{s} ***· (1 + x)*** ^{q} **Ξ*(1 + x*** ^{p} ***)*** ^{s} ***· (1 + x)*** ^{q} ***(mod p***

$$\equiv \sum_{i=0}^{s} \binom{s}{i} \cdot x^{i \cdot p} \cdot \sum_{j=0}^{q} \binom{q}{j} \cdot x^{j} \quad \text{(modp)}$$

所以得***(1 + x)*** $^{sp+q}$ $\equiv \sum_{i=0}^{s} \binom{s}{i} \cdot x^{i \cdot p} \cdot \sum_{j=0}^{q} \binom{q}{j} \cdot x^{j}$ **(mod p)**

我们求左边***(1 + x)*** $^{sp+q}$ 中的 x^{tp+r} 的系数为：

$$\binom{sp+q}{tp+r}$$

求右边公式中的 x^{tp+r} 为：

通过观察你会发现当且仅当i = t , j = r ,能够得到 x^{tp+r} 的系数，即

$$\binom{s}{t}\binom{q}{r}$$

所以

$$\binom{sp+q}{tp+r} \equiv \binom{s}{t}\binom{q}{r}\,(mod\,(p))$$

得证。

预处理1到p的阶乘即可，总时间复杂度：$O(p+log_p n \times p)$

```
x
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
const int N=1e5+5,M=1e6+5,inf=0x3f3f3f3f,mod=1e9+7;
#define mst(a) memset(a,0,sizeof a)
#define lx x<<1
#define rx x<<1|1
#define reg register
#define PII pair<int,int>
#define fi first
#define se second
ll ksm(ll a,ll n,ll p){
    ll ans=1;
    while(n){
        if(n&1) ans=ans*a%p;
        a=a*a%p;
        n>>=1;
    }
    return ans;
}
ll fac[N];
int p;
ll C(ll n,ll m){    //组合数&费马小定理.
    if(n<m) return 0;
    return fac[n]*ksm(fac[m]*fac[n-m]%p,p-2,p)%p;
}
ll Lucas(ll n,ll m){    //递归求解.
    if(!m) return 1;
    return C(n%p,m%p)*Lucas(n/p,m/p)%p;
}
int main(){
    int t;
    scanf("%d",&t);
    while(t--){
        int n,m;
        scanf("%d%d%d",&n,&m,&p);
        fac[0]=1;
        for(int i=1;i<=p;i++) fac[i]=fac[i-1]*i%p;
        printf("%lld\n",Lucas(n+m,n));
    }
    return 0;
}
```