

原创

【Luogu 3807】Lucas定理

2019-08-28 16:25:09

我是一只计算鸡

阅读数 21

更多

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。
本文链接：<https://blog.csdn.net/giftedpanda/article/details/100121738>

给定 n, m, p (p 为素数)，求 $\binom{n+m}{m} \% p$

Lucas定理: $Lucas(n, m, p) = C(n \% p, m \% p) * Lucas(n / p, m / p, p)$

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 const int maxn = 1e5 + 7;
4 typedef long long ll;
5 ll fact[maxn], inv[maxn];
6 ll n, m, p;
7 ll power(ll a, ll b, ll mod) // 快速幂 费马小定理求逆元
8 {
9     a %= mod;
10    ll ans = 1;
11    while(b) {
12        if(b & 1) ans = (ans * a) % mod;
13        a *= a;
14        a %= mod;
15        b >>= 1;
16    }
17    return ans;
18 }
19 ll C(ll n, ll m, ll p) // 组合数
20 {
21     if(n < m) return 0;
22     return fact[n] * inv[n-m] % p * inv[m] % p;
23 }
24 int Lucas(ll n, ll m, ll p) // Lucas定理
25 {
26     if(m == 0) return 1;
27     return C(n % p, m % p, p) * Lucas(n / p, m / p, p) % p;
28 }
29 int main()
30 {
31     int T;
32     scanf("%d", &T);
33     while(T--) {
34         scanf("%lld %lld %lld", &n, &m, &p);
35         fact[0] = fact[1] = inv[0] = inv[1] = 1;
36         for(int i = 2; i <= p; i++) fact[i] = (fact[i-1] * i) % p; // 处理阶乘
37         inv[p-1] = power(fact[p-1], p-2, p);
38         for(int i = p-2; i > 1; i--) inv[i] = (inv[i+1] * (i + 1)) % p; // 递推求阶乘逆元
39         printf("%lld\n", Lucas(n+m, m, p));
40     }
41     return 0;
42 }
```

有 0 个人打赏

文章最后发布于: 2019-08-28 16:25:09