二项式反演

算法简介

一种特殊反演，主要应用于组合计数。

算法思想

$$f(n)=\sum_{i=m}^n{n\choose i}g(i)\iff g(n)=\sum_{i=m}^n(-1)^{n-i}{n\choose i}f(i)$$

$$f(n)=\sum_{i=n}^m{i\choose n}g(i)\iff g(n)=\sum_{i=n}^m(-1)^{i-n}{i\choose n}f(i)$$

算法例题

题意

一个有 n 个元素的集合，现在要从他的所有子集中取出至少一个集合，使得所选子集的交集的元素个数为 k，求方案数。

题解

定义 $f(k)$ 表示所有钦定 k 个元素属于交集(其余元素任意)的方案数之和，于是有 $f(k)={n\choose k}2^{2^{n-k}}$。

定义 $g(k)$ 表示恰好有 k 个元素属于交集的方案数，对于 $f(k)$ 的每个方案，$g(i)(i\ge k)$ 被计算 ${i\choose k}$ 次，于是 $f(k)=\sum_{i=k}^n{i\choose k}g(i)$。

根据二项式反演，有 $g(k)=\sum_{i=k}^n(-1)^{i-k}{i\choose k}f(i)$，时间复杂度 $O(n)$。

算法练习

习题一

洛谷p5505

题意

有 n 个人和 m 种物品，第 i 种物品有 a_i 个，同种物品之间没有区别。现在要将这些物品分给这些人，使得每个人至少分到一个物品，求方案数。

题解

定义 $f(k)$ 表示所有钦定 k 个人没有分到物品(其余人任意)的方案数之和，于是有 $f(k)={n\choose k}\prod_{i=1}^m{n-k+a_i-1\choose n-k-1}$

定义 $g(k)$ 表示恰好有 k 个人没有分到物品的方案数，对于 $f(k)$ 的每个方案$g(i)(i\ge k)$ 被计算 ${i\choose k}$ 次，于是 $f(k)=\sum_{i=k}^n{i\choose k}g(i)$

根据二项式反演，有 $g(k)=\sum_{i=k}^n(-1)^{i-k}{i\choose k}f(i)$题目所求即为 $g(0)$时间复杂度 $O(nm)$

```
const int MAXN=2005,Mod=1e9+7;
int a[MAXN],C[MAXN][MAXN];
int main()
{
    int n=read_int(),m=read_int();
    _for(i,0,m)a[i]=read_int();
    C[0][0]=1;
    _for(i,1,MAXN){
        C[i][0]=1;
        _rep(j,1,i)
        C[i][j]=(C[i-1][j-1]+C[i-1][j])%Mod;
    }
    int ans=0;
    _rep(i,0,n){
        int t=C[n][i];
        _for(j,0,m)t=1LL*t*C[n-i+a[j]-1][n-i-1]%Mod;
        if(i&1)ans=(ans-t)%Mod;
        else ans=(ans+t)%Mod;
    }
    enter((ans+Mod)%Mod);
    return 0;
}
```

习题二

洛谷p4859

题意

题解

```
```