2020牛客暑期多校训练营（第五场）C Easy

题意

C Easy
有两个长度为k的正整数序列A，B满足$\sum_{i=1}^{k} a_i = N$,$\sum_{i=1}^{k} b_i = M$
求$P = \prod_{i=1}^{k} min(a_i, b_i)$的种类。

题解

设$n < m$，满足条件的a和b的方案构造的生成函数如下

$(x + x^2 \ldots + \ldots + x^n)^k (y^1 + y^2 \ldots + \ldots + y^m)^k$

其中x, y的幂次表示给予的值，所以$(x + x^2 \ldots + \ldots + x^n)^k$拆开后的$x^n$项的系数就表示$a_i$的种类数，y同理。

因此可知$x^n y^m$的系数即为赋值所求方案数。
然后构造题中$min(a_i, b_i)$的值，
单一的一对(a_i, b_i)其生成函数为$(x + x^2 \ldots + \ldots + x^n)(y^1 + y^2 \ldots + \ldots + y^m)$,那么此时该项的贡献价值应为$min(a_i, b_i)$，显然该生成函数中每一项的系数都只有1。为了构造正确贡献的生成函数，使得每一对的(a_i, b_i)都有$min(a_i, b_i)$的系数，因此对于每一项$x^i y^j$（$i < j$）,考虑$x^{i-1} y^{j-1}$,$x^{i-2} y^{j-2} \ldots x^1 y^{j-i+1}$项，每一项都乘上$xy$的某个幂次，就可以使每一项都具有$min(a_i, b_i)$的系数。因此对于单一的一项，构造如下生成函数
$(x + x^2 \ldots + \ldots + x^n)^k (y^1 + y^2 \ldots + \ldots + y^m)^k (1 + xy + (xy)^1 + (xy)^2 + \ldots + (xy)^{n-k})$
再扩展到k项为如下生成函数
$(x + x^2 \ldots + \ldots + x^n)^k (y^1 + y^2 \ldots + \ldots + y^m)^k (1 + xy + (xy)^1 + (xy)^2 + \ldots + (xy)^{n-k})^k$
这个式子通过公式$G(x) = \frac{1}{1-x} = x + x^2 \ldots + \ldots + x^n$就可以转化为题解中的式子$\frac{xy}{(1-x)(1-y)(1-xy)}$（所以题解为什么要转过来啊orz）
在生成函数中构造$x^n y^m$就可以枚举xy的系数$i = [0, n-k]$（因为序列要是正整数，所以先减去k个数），然后补全x，y的系数，其中通过广义二项式$G^n(x) = \sum_{k=0}^{\infty} C_{n+k-1}^{k} x^k$扩展。公式如下：$res = \sum_{i=0}^{n-k} C_{k+i-1}^{i} C_{k+n-k-i-1}^{n-k-i} C_{k+m-k-i-1}^{m-k-i}$

代码

```
#include<bits/stdc++.h>
typedef long long ll;
using namespace std;
#define N 1000005
#define mod 998244353
```

```
ll powmod(ll a, ll b){
    ll res = 1;
    a %= mod;
    while(b) {
        if(b & 1) {
            res = res * a % mod;
        }
        b >>= 1;
        a = a * a % mod;
    }
    return res;
}
ll fac[N], inv[N];
void init() {
    fac[0] = 1;
    for(int i = 1; i < N; i++) {
        fac[i] = fac[i - 1] * i % mod;
    }
    inv[N - 1] = powmod(fac[N - 1], mod - 2);
    for(int i = N - 2; i >= 0; i--) {
        inv[i] = inv[i + 1] * (i + 1) % mod;
    }
}
ll c(ll n, ll m) {
    if(m > n) {
        return 0;
    }
    if(m == 0) return 1;
    if(n < N) return fac[n] * inv[m] % mod * inv[n - m] % mod;
    //c(m,k) = m!/(k! * (m - k))= m-k+1~m/(k!)
    ll res = inv[m];
    for(int i = n - m + 1; i <= n; i++) {
        res = res * i % mod;
    }
    return res;
}
int main() {
    init();
    int t;
    scanf("%d", &t);
    while(t--) {
        int n, m, k;
        scanf("%d%d%d", &n, &m, &k);
        if(n > m) swap(n, m);
        ll res = 0;
        for(int i = 0; i <= n - k; i++) {
            res = ((ll)res + c(k + i - 1, i)
                        * c(k + n - k - i - 1, n - k - i) % mod
                        * c(k + m - k - i - 1, m - k - i) % mod) % mod;
        }
        printf("%lld\n", res);
    }
}
```