G. Ball

题意

给定一个斜坡，有 n 个洞。再给定 m 个球，依次抛球，每次抛球可以决定球的初始下落位置，然后球从斜坡向下运动。

如果球遇到空洞则将该洞填补，否则向下一个洞运动。如果球没有遇到任何空洞，则出界。问恰好出界 k 个球的方案数。

题解

设 $f(i,j)$ 表示 i 个洞投 j 个球且没有球出界的方案数，$g(i,j)$ 表示 i 个洞投 j 个球且所有洞都被填满的方案数。

枚举终态时从斜坡自下向上第一个空洞的位置 i，于是斜坡被分成两段，上段斜坡 $[i+1,n]$ 所有球一定不能出界，否则位置 i 将不是空洞。

下段斜坡 $[1,i-1]$ 一定全部被填满，且为保证有 k 个球出界，一定恰好有 $i-1+k$ 个球投向下段斜坡。

$$ \text{ans}\gets \sum_{i=1}^{n} g(i-1,i-1+k)f(n-i,m-i-k+1){m\choose i-1+k} $$

还要考虑终态没有空洞的情况

$$ \text{ans}\gets [m=n+k]g(n,m) $$

接下来考虑计算 $f(i,j),g(i,j)$，对 $f(i,j)(i\ge j)$，可以枚举有 k 个球被抛向位置 i，不难发现剩下 $j-k$ 个球对应方案 $f(i-1,j-k)$。

证明：不难发现交换投球顺序不影响终态，于是不妨假设这 k 个球是最后投的。

由于前 $j-k$ 个球投完剩下 $i-j+k\ge k$ 个洞，于是 k 个球可以全部进洞，证毕。最终有

$$ f(i,j)=\sum_{k=0}^j f(i-1,j-k){j\choose k} $$

对 $g(i,j)(i\le j)$，可以枚举最后一个球的进洞位置，同样可以把斜坡分两段，顺便考虑一下最后一个球出界的情况，于是有

$$ g(i,j)=\sum_{k=0}^{i-1}\left(f(k,k)g(i-k-1,j-k-1){j-1\choose k}(k+1)\right)+g(i,j-1)i $$

于是可以 $O\left(n^2m\right)$ 预处理，$O(n)$ 处理每个询问。

```
const int MAXN=505,MAXM=1e3+5,mod=998244353;
int f[MAXN][MAXM],g[MAXN][MAXM],C[MAXM][MAXM];
void Init(){
    C[0][0]=1;
    _for(i,1,MAXM){
        C[i][0]=1;
        _rep(j,1,i)
        C[i][j]=(C[i-1][j-1]+C[i-1][j])%mod;
```

```
    }
    f[0][0]=g[0][0]=1;
    _for(i,1,MAXN){
        f[i][0]=1;
        _rep(j,1,i){
            _rep(k,0,j)
            f[i][j]=(f[i][j]+1LL*C[j][k]*f[i-1][j-k])%mod;
        }
        _for(j,i,MAXM){
            _for(k,0,i)
            g[i][j]=(g[i][j]+1LL*f[k][k]*g[i-k-1][j-
k-1]%mod*C[j-1][k]%mod*(k+1))%mod;
            g[i][j]=(g[i][j]+1LL*g[i][j-1]*i)%mod;
        }
    }
}
int main(){
    Init();
    int T=read_int();
    while(T--){
        int n=read_int(),m=read_int(),k=read_int(),ans=0;
        for(int i=0,t=min(n,m-k+1);i<t;i++)
        ans=(ans+1LL*g[i][i+k]*f[n-i-1][m-i-k]%mod*C[m][i+k])%mod;
        if(m==n+k)
        ans=(ans+g[n][m])%mod;
        enter(ans);
    }
    return 0;
}
```