直接考虑正解吧。

我们定义$f_{i,x}=\sum_{j=1}^{n}C_{tj+i}^{x}$。

那么对于每次询问x，要的其实就是$f_{0,x}$。

现在我们知道：

$\sum_{i=0}^{t-1}f_{i,x}=\sum_{i=t}^{tn}C_{i}^{x}$

以及一个组合恒等式：

$\sum_{i=1}^{tn}C_{i}^{x}=C_{tn+1}^{x+1}$。（式子1）

以及另外一个简单推理：

$f_{i,x}=\sum_{j=1}^{n}C_{tj+i}^{x}$

$f_{i-1,x-1}=\sum_{j=1}^{n}C_{tj+i-1}^{x}$

$f_{i-1,x}=\sum_{j=1}^{n}C_{tj+i-1}^{x-1}$

显然的：

$f_{i,x}=f_{i-1,x-1}+f_{i-1,x}$（式子2）

于是，我们可以考虑按x从小到大，依次求出$f_{i,x}$。

具体怎么求呢?

当我们要求$f_{i,x}$的时候，我们假设$f_{i,x-1}$已经求出来了。

那么：我可以根据式子1和式子2，列一个t元一次方程组来高斯消元，这样对于每一个x，复杂度是t^3的。总复杂度是$nt\times t^3$的。

反正$t=3$是完全可以做出来的。

怎么求正解呢？答案就是代入消元了。

我们用$f_{0,x}$来表示$f_{1,x}$，再继续用$f_{0,x}$来表示$f_{2,x}$，...，这么表示完以后，把他们加起来，就变成了$af_{0,x}+b=\sum_{i=t}^{tn}C_i^x$了，就可以解出$f_{0,x}$以及其他变量了。

复杂度$O(nt^2)$。给5秒是因为我写的有点屎，懒得卡常，以及要给n^3高斯消元的分数（虽然我感觉没什么人能想到$O(t^3)$但想不到$O(t)$。

```
for (int i=0;i<t;i++) f[0][i]=n;
for (int i=1;i<=t*n;i++)
{
    for (int j=0;j<=t;j++) memset(tmp[j].v,0,sizeof(tmp[j].v));
    tmp[0].v[0]=1;
    for (int j=1;j<t;j++)
    {
        add(tmp[j],tmp[j-1]);
        add(tmp[j].v[1],f[i-1][j-1]);
    }
    for (int j=1;j<t;j++) add(tmp[0],tmp[j]);

    ll tot=cc(t*n+t,i+1);
    for (int j=1;j<t;j++) tot=tot-cc(j,i);
    tot=tot-tmp[0].v[1];
    tot=(tot%mod+mod)%mod;
    ll ni=pw(tmp[0].v[0],mod-2);
    f[i][0]=ni*tot%mod;

    for (int j=1;j<t;j++)
    {
        f[i][j]=((f[i][0]*tmp[j].v[0]+tmp[j].v[1])%mod+mod)%mod;
    }
}
```