根据异或的性质,X_0⊕X_1⊕X_2⊕X*3⊕…..⊕X*(M-1)=S⊕T 计算出S⊕T中一共有cnt个1，将这个二进制数标准化,使它成为前面N-cnt个0，后cnt个1的二进制数，答案保持不变，问题变为:用K个N位且有3个1的不同二进制数进行异或，最终得到前面N-cnt个0，后面cnt个1这个二进制数，有多少种方案？ 递推。设d[K][M]表示用K个N位且有3个1的不同二进制数进行异或，最终得到前面N-cnt个0，后面cnt个1这个二进制数的方案数。 递推方程： d[i][j] = d[i-1][j+1] *C[j][1]* C[n-j][2]+ d[i-1][j+3] * C[n-j][3]

- d[i-1][j-1] *C[j][2]* C[n-j][1] 此时得到的d[i][j]未去除加入数字的先后顺序和加入重复串带来的影响 考虑加入重复串的影响，d[i][j]-= (d[i-2][j] *(C[n][3]-i+2);表示去除加入i-2个数字并且后面有j个1这个二进制数但后来又异或上两个之前没用过的相同的数的方案数， 考虑二进制数加入的先后顺序给答案带来的额外贡献，d[i][j]=d[i][j]*inv[i]%mod 即d[i][j]/i，其中inv[i]为数字i关于模mod的逆元