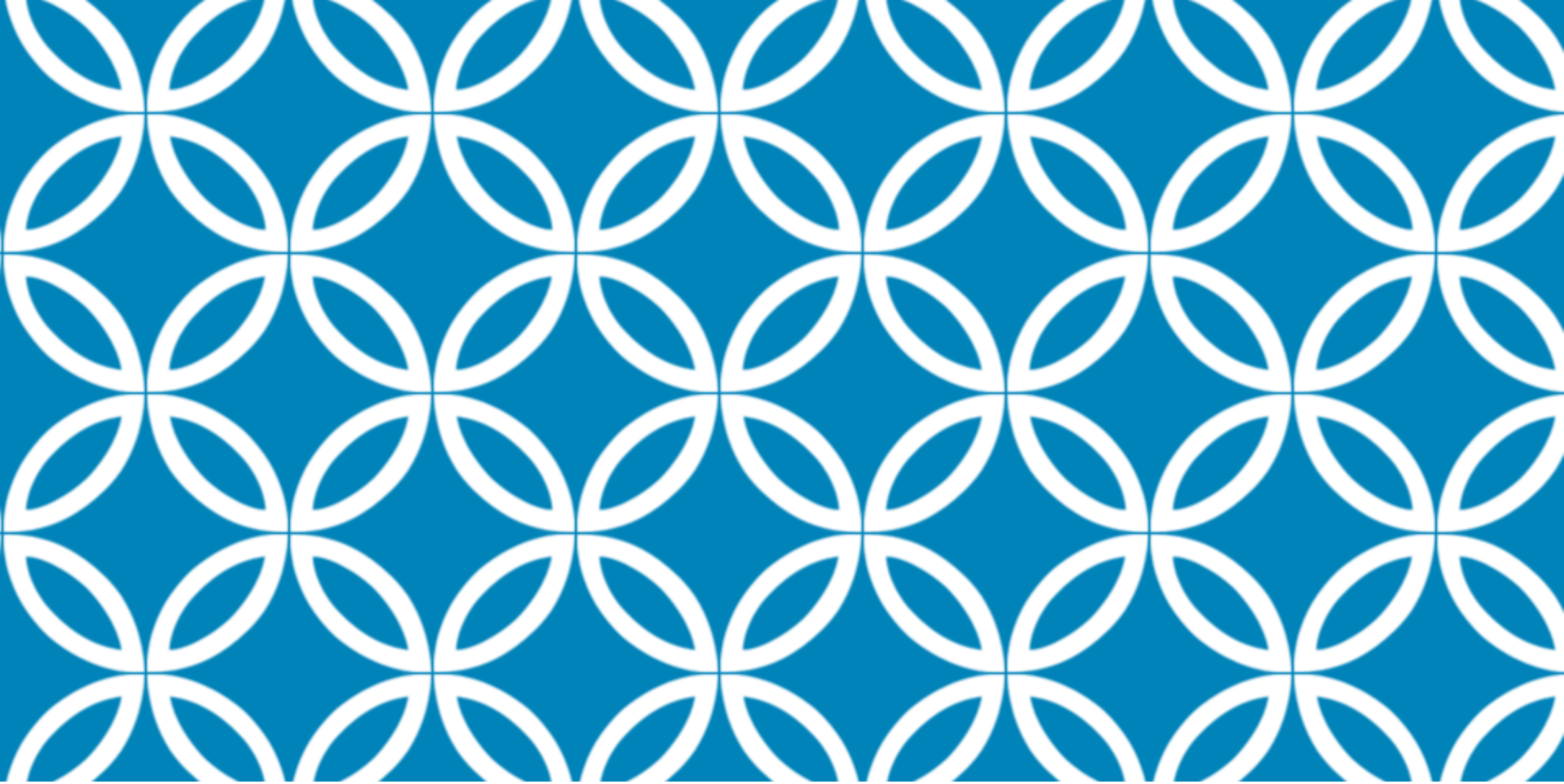


TopCoder 题目选讲

湖南师大附中 胡泽聪

介绍

TC大家都熟悉吧，就不介绍了。

剧透：下面要讲的题以网络流和组合数学为主。

TCO 2013 Round 1A D1L3 DirectionBoard

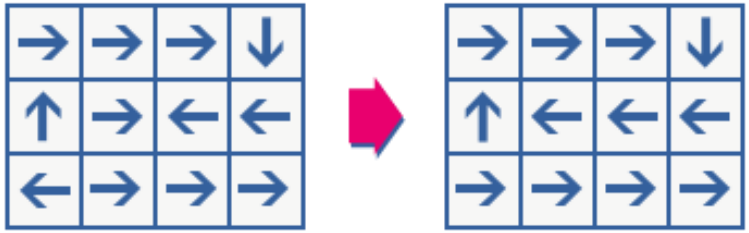
一块$N \times M$的地图，每个格子内有一个箭头。

从某个格子出发，沿着箭头的方向走，从一侧出了边界就从另一侧进入。

如果从任意一个格子出发都能回到出发的格子，那么地图就是合法的。

给定地图，求：至少要改变几个格子的箭头的方向，才能使得地图合法。

$N, M \leq 15$。



TCO 2013 Round 1A D1L3 DirectionBoard

先考虑判断一个地图是否合法。

建立图论模型。

把每个格子抽象成节点，向箭头指向的格子连边。

每个格子可以沿箭头回到自己⇔图论模型中的每个节点都在一个环中

而每个节点的出度为1，所以即图中每个连通块都是环。

也即每个点的入度为1，出度为1。

联想到了什么?

流量平衡!

TCO 2013 Round 1A D1L3 DirectionBoard

建立网络流模型。

一个格子拆成两个点，称之为出点和入点。

从源向出点、从入点向汇连一条容量为1的边。

每个格子的出点向箭头指向的格子的入点连一条容量为1的边。

满流即合法。

TCO 2013 Round 1A D1L3 DirectionBoard

如果要改方向呢?

将网络流模型改造为费用流模型。

原有的边费用为0。

每个格子的出点向相邻的非箭头所指格子的入点连容量为1，费用为1的边。

答案即最小费用最大流。

SRM 570 D1L3
CurvyonRails

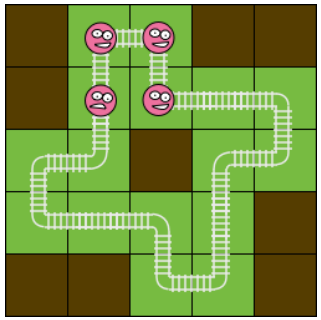
一块$N \times N$的地图，每个格子均为空地或者障碍。

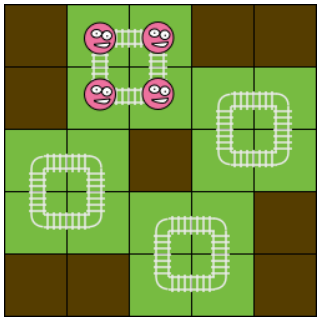
现在要铺铁轨，每块空地都要被覆盖，每条路线必须闭合，且不能相交。

有些空地上住着弯星人，在这种空地上铺一块直的铁路需要花费1的代价。

给定地图，求：是否能够铺铁轨，以及最小代价。

$N \leq 25$。





SRM 570 D1L3
CurvyonRails

第一眼看上去……

插头DP?

$N \leq 25$，TLE。

这题和需要用插头DP的题有什么区别?

需要用插头DP的题对所求路径有特殊要求（比如为哈密尔顿回路），而且有些题需要求方案数。

而此题只需覆盖每个格子，且所求为某个最小值。

或许……

网络流?

SRM 570 D1L3 CurvyonRails

先考虑判断是否有解。

有解⇔存在一些不相交不重复的回路可以覆盖所有空地且不覆盖任意障碍

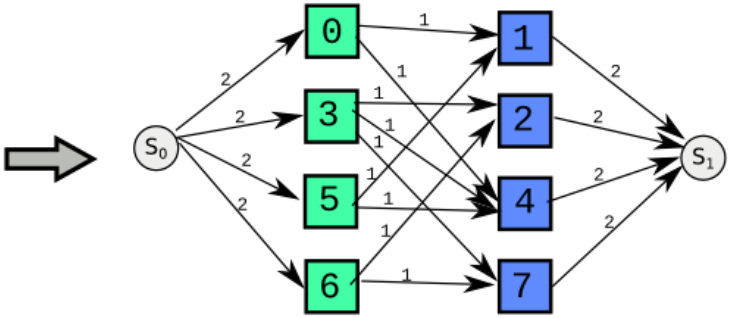
也即，每个空地向相邻的空地连出恰好两条轨道。

那么构建网络流模型。

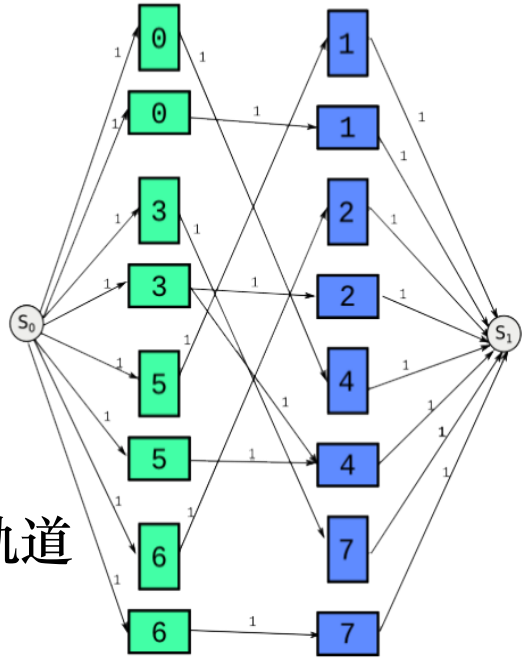
对地图（只考虑空地）黑白染色，从源向黑点、从白点向汇连容量为2的边。

从黑点向相邻的白点连容量为1的边。

满流即有解。



SRM 570 D1L3 CurvyonRails



那么弯的和直的铁轨怎么判断?

直的铁轨⇔一块空地连出两条横向或纵向的轨道

弯的铁轨⇔一块空地连出一条横向和一条纵向的轨道

先强制所有铁轨都是弯的，判断是否有解。

在已有的网络流模型基础上改进。

把每块空地拆成两个点，一个代表横向，一个代表纵向。

横向点向横向相邻的空地的横向点连一条容量为1的边，纵向的类似。

满流即有解。

SRM 570 D1L3
CurvyonRails

如果要把一块铁轨改成直的呢?

如果是没人的空地，随便改；如果是有人的空地，收取费用。

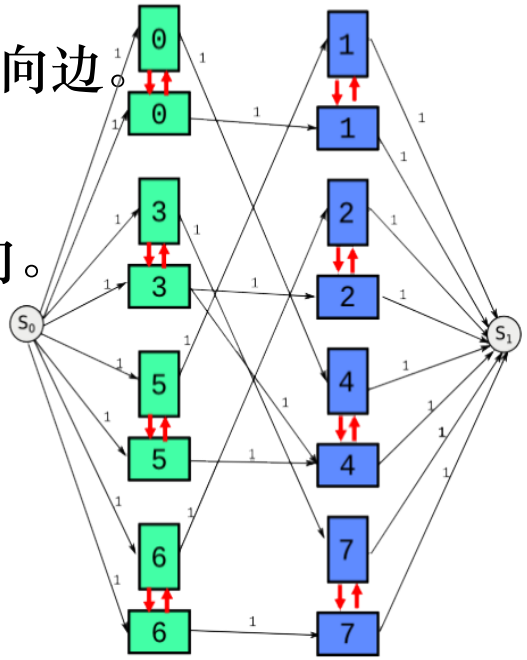
把网络流模型改成费用流模型，已有的边费用为0。

在一块空地拆出的两个点之间连一条流量为1的双向边。

如果空地有人，双向边的费用为1；否则为0。

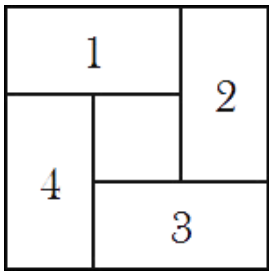
如果这类边有流量，就说明有一条轨道改变了方向。

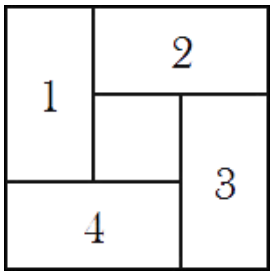
答案即为最小费用最大流。



SRM 521 D1L3
Chimney

一个烟囱共N层，每层由4块砖头构成，奇数层和偶数层的砖头摆放分别如图：





摆放非底层的一块砖头时，只有其下方的两块砖头都已经摆放好了的时候，才可以摆放。

求摆放所有砖头的顺序的方案数。

$N \leq 10^{18}$。

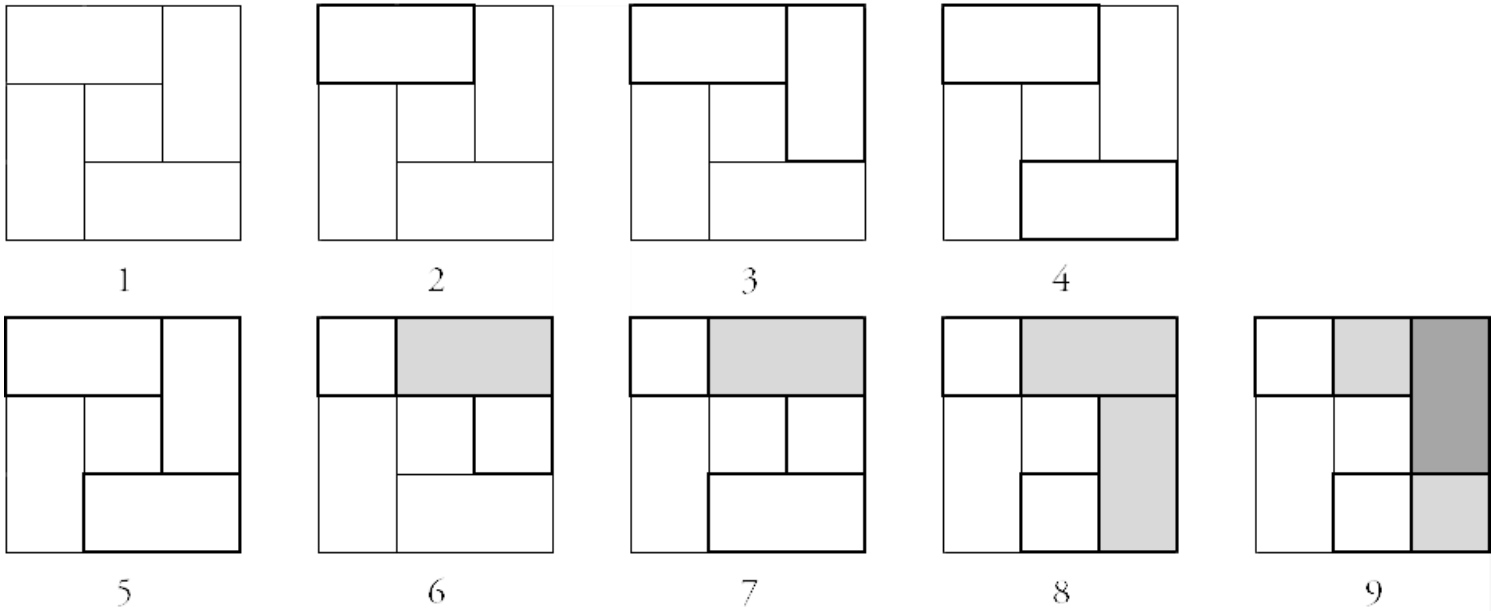
SRM 521 D1L3
Chimney

看到这数据范围……矩阵乘法?

那么先考虑状态表示。

假设已经摆完了前i层的砖头，此时会有多少种状态?

一共9种（以奇数层的为例）：



SRM 521 D1L3
Chimney

据此构造出的矩阵如下：

$$\begin{bmatrix} 24 & 64 & 64 & 64 & 0 & 0 & 0 & 0 & 0 \\ 6 & 16 & 16 & 16 & 0 & 0 & 0 & 0 & 0 \\ 2 & 6 & 6 & 6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 2 & 0 & 0 & 0 & 0 & 0 \\ 2 & 4 & 4 & 4 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 2 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

计算矩乘快速幂即可。

答案即最左上角的元素的值。

SRM 472 D1L2
TwoSidedCards

有N张牌，正反都有数字。

所有正面和所有反面的数字各构成一个排列。

将牌排成一排，每张牌可以正面朝上或者反面朝上，这样可以构成一个序列。

给定N张牌正面和反面的数字，求：所有可能的序列方案数。
$N \leq 50$。

比如：3张牌，正面分别为1、2、3，反面分别为1、3、2。

一共有12种序列：

123、132、213、231、312、321、
133、313、331、122、212、221。

SRM 472 D1L2 TwoSidedCards

假设N足够小，可以让我们枚举哪些牌正面朝上。

那么会发现，可能会有一些数字重复出现，而且重复出现次数最多为2。

记重复出现的数字个数为K，那么这些牌的不同排列数为：

$$\frac{N!}{2^K}$$

即多重集的排列数。

SRM 472 D1L2 TwoSidedCards

现在我们要求的就是，对于任意合法的K，在多少种牌的正反方案中有K个数出现了2次。

我们把所有牌按正面数值1到N的顺序摆成一排。

这样我们就可以把反面数值视为一个置换，把这张牌翻面就相当于置换这一个位置。

可以发现，置换的不同循环之间是不影响的。

现在我们就来考虑一个循环。

SRM 472 D1L2 TwoSidedCards

记这个循环的长度为L，假设出现两次的数的个数为K。

显然有$K \leq L/2$。

……然后呢？似乎不太好求啊。

尝试转换思路，比如……

建立图论模型。

将循环的每个元素视为一个节点，每个元素连向元素的值对应的元素的节点。

SRM 472 D1L2 TwoSidedCards

有了这个图可以干什么呢?

一张牌可以视为其中的一条边，牌是否反面朝上相当于边是否反向。牌的数字也即边指向的节点对应元素的值。

那么出现两次的数也即入度为2的节点对应元素。

而且还可以发现，入度为2的节点和入度为0的节点个数相等，并且交替出现。

那么我们就可以得出一个计算长度为L的循环中有K个数出现两次的方案数的公式:

$$2 \times \binom{L}{2K}$$

$K = 0$时方案数为1。

SRM 472 D1L2 TwoSidedCards

注意刚才只考虑了牌的正反，还没有考虑排列。

剩下的部分就不难了。

可以用背包求出对于整个序列出现两次的数为K个的方案数，再用多重集的排列公式计算求和。

也可以处理出每个循环内部出现两次的数为任意个的方案数，然后把一个循环视为同一种元素，还是一个多重集排列的问题。

SRM 473 D1L3
RooksParty

有N种颜色的车（国际象棋），每种Col_i个，放置在$R \times C$的棋盘上。

如果有两个不同颜色的车处于同行或同列，且之间没有其他车，它们就会互相攻击。

求使得所有车不互相攻击的摆放方案数。

$R, C \leq 30$、$N \leq 10$。

SRM 473 D1L3
RooksParty

每行和每列的所有格子要么是同一种颜色的车，要么为空。

给每行和每列标上一种颜色。

一种颜色的车能放置的格子为，该行和该列的颜色都为车的颜色的格子。也即每种颜色之间无关。

令$F[col][x][y]$表示颜色$1\sim col$的车占据x行和y列的方案数。

$$F[col][x][y]=\sum_{r=1}^{x}\sum_{c=1}^{y}F[col-1][x-r][y-c]\cdot\binom{x}{r}\cdot\binom{y}{c}\cdot S(Col_c,r,c)$$

其中$S(n,r,c)$为将n个车摆放在$r\times c$的棋盘上，使得每行每列都有至少一个车的方案数（不考虑互相攻击，因为同色嘛）。

注意边界和转移的合法性。

SRM 473 D1L3 RooksParty

那么怎么求$S(n,r,c)$?

先只考虑列。令$R(n,r,c)$代表只考虑列的方案数，运用容斥原理:

$$R(n,r,c)=\sum_{i=0}^{c}(-1)^i\binom{r(c-i)}{n}\binom{c}{i}$$

其中i代表有i列为空。

然后再把行加入考虑

$$S(n,r,c)=\sum_{j=0}^{r}(-1)^j R(n,r-j,c)\binom{r}{j}$$

其中j代表有j行为空。

SRM 473 D1L3
RooksParty

还有另一种DP的方法。

令$G[n][r][c]$表示n个车放在$r \times c$的棋盘上的方案数。

假设不考虑每行每列都有车的条件，那么方案数为$\binom{r \cdot c}{n}$。

此时再减去有某些行和某些列为空的方案，即：

$$G[n][r][c] = \binom{r \cdot c}{n} - \sum_{i=1}^{r} \sum_{j=1}^{c} \binom{r}{i} \binom{c}{j} G[n][r-i][c-j]$$

同样，注意边界和转移的合法性。

SRM 473 D1L3
RooksParty

让我们来分析时间复杂度。

计算F数组的复杂度：

状态数：$O(N \cdot R \cdot C)$；转移：$O(R \cdot C)$。

复杂度$O(NR^2C^2)$。

基于容斥原理的算法：

状态数：$O(maxN \cdot R \cdot C)$；转移：$O(R + C)$。

但注意到$S(n, r, c)$只和$S(n, \ldots, \ldots)$的状态有关，故状态数为$O(N \cdot R \cdot C)$。

故复杂度为$O\big(NRC(R + C)\big)$，总复杂度$O(NR^2C^2)$。

SRM 473 D1L3
RooksParty

计算G数组的复杂度：

状态数：$O(N \cdot R \cdot C)$（和容斥原理类似）；转移：$O(R \cdot C)$。

复杂度$O(NR^2C^2)$。

观察式子：

$$G[n][r][c] = \binom{r \cdot c}{n} - \sum_{i=1}^{r}\sum_{j=1}^{c}\binom{r}{i}\binom{c}{j}G[n][r-i][c-j]$$

稍作修改：

$$G[n][r][c] = \binom{r \cdot c}{n} - \sum_{i=1}^{r}\binom{r}{r-i}\sum_{j=1}^{c}\binom{c}{c-j}G[n][r-i][c-j]$$

维护$\sum_{j=1}^{c}\binom{c}{c-j}G[n][r-i][c-j]$，可以优化到$O(NR^2C)$。

SRM 473 D1L3
RooksParty

现在瓶颈在于F数组的计算。

事实上用一些很牛逼的方法可以优化到大概$O(NRC \log R \log C)$的复杂度，有兴趣的同学可以自行研究。/*（其实是我不会……）*/

当然，即使不优化，对于本题来说也是可以通过的。

SRM 573 D1L3
WolfPack

在一个无限的平面上有N匹狼。每匹狼都要走M步。

一匹狼一步可以移动到与其所在格子四连通的另一个格子。同一个格子可以被一匹狼经过多次，一个格子上也可以有多只狼。

在M步之后，所有的狼都要汇聚于同一个格子。

给定所有狼的初始坐标和M，求：汇聚的方案数。两个汇聚方案不同当且仅当存在一匹狼在两个方案中某步的移动方向不同。

$N \leq 50$，$M \leq 100{,}000$。

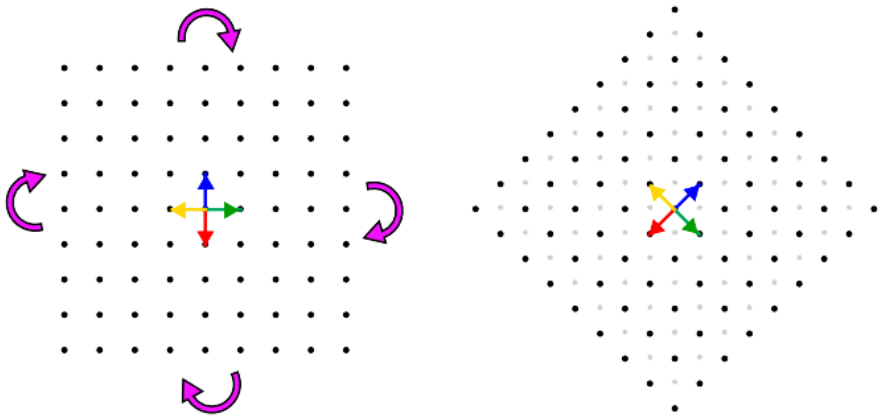
SRM 573 D1L3 WolfPack

事实上这题有一个很好的性质：

将平面顺时针旋转45度，一匹狼横向移动的距离和纵向移动的距离是无关的。

~~意识流~~证明如下：

假设旋转前坐标为(x, y)，旋转之后就变成了$(x+y, x-y)$。旋转前一匹狼可以向上下左右移动，旋转之后就变成了向右上、左下、左上、右下移动。这样横向移动时纵向可以向两个方向移动，纵向移动时同理。



SRM 573 D1L3
WolfPack

那么我们按横纵坐标分别计算。

注意到M不超过100,000，所以我们可以枚举汇聚位置。

如果有一匹狼所在的格子和汇聚位置的颜色不同，那么这个位置无效。

否则计算每匹狼到汇聚位置的距离，记为D_i。方案数即：

$$\prod \binom{M}{\frac{M-D_i}{2}}$$

所有有效汇聚位置的方案数之和即该维坐标的方案数。两维坐标的方案数的乘积即为答案。

预处理组合数即可。

TCO09 Championship Round D1L2 ArrayTransformations

给定序列A，记A的元素个数为N。

称一次变换为：选定区间$[i,j]$，满足$1 \le i \le j \le N$，对序列A在区间中的每个数减1，如果已经为0则不操作。这样一次变换的代价为$j-i+1$。

对于给定序列A，最大变换次数K和最大代价和M，求：在使用不超过K次变换，总变换代价不超过M的前提下，A中最大元素的最小值是多少。

$N \le 250$。

比如：$A=\{1,5,9,2,9,0,1\}$，$K=5$，$M=10$。

一种方案是对区间[2,6]、[3,3]、[3,3]、[5,5]、[5,5]应用变换。

变换总代价为9，应用变换之后A中最大元素为6。

TCO09 Championship Round D1L2 ArrayTransformations

……似乎不好下手？

考虑二分答案，把问题转换成判定性问题。

现在的问题是，能否在限制下使A中最大元素不超过一个值P。

容易发现，变换的顺序并不影响最后的结果。

对于$A[i]$，变换的最小代价应为$\max(A[i]-P,0)$。那么我们令$B[i]=\max(A[i]-P,0)$。

那么限制条件实际上就是要确定不超过K条线段，使得数轴上i的位置被覆盖了至少$B[i]$次，且线段总长不超过M。

根据此我们来建立费用流模型（我也觉得这里是神转折）。

TCO09 Championship Round D1L2 ArrayTransformations

建立标号为1到$N+1$的点，以及S'和T'节点。

从源向S'连一条容量为K的边、从T'向汇连一条容量为K、费用为0的边。

从标号为i的点向标号为$i+1$的点连一条容量为∞、容量下界为$B[i]$、费用为1的边。

从S'向1~N号节点、从2~$N+1$号节点向T'连一条容量为∞、费用为0的边。

一次变换，即一条线段$[i,j]$，在网络中会是源$\rightarrow S' \rightarrow i \rightarrow i+1 \rightarrow \cdots \rightarrow j \rightarrow T' \rightarrow$汇的一条增广路。

如果最小费用最大可行流得到的费用不超过M，那么答案合法。

TCO09 Championship Round D1L2 ArrayTransformations

怎么求最小费用最大可行流?

……

怎么求带下界的可行流?

怎么求最大可行流?

TCO09 Championship Round D1L2 ArrayTransformations

当然费用流不是唯一的做法。

这题还有二分答案后贪心判断可行的方法。详见官方题解。