

没有什么特殊难题，考察搜索的应用，包括退火算法

Translate:USACO/snail

Snail Trails 蜗牛的旅行

All Ireland Contest

译 by Felicia Crazy

描述

萨丽·斯内尔 (Sally Snail, 蜗牛) 喜欢在 $N \times N$ 的棋盘上闲逛 ($1 \leq n \leq 120$)。

她总是从棋盘的左上角出发。棋盘上有空的格子 (用 "." 来表示) 和 B 个路障 (用 "#" 来表示)。下面是这种表示法的示例棋盘：

	A	B	C	D	E	F	G	H
1	S	#	.
2	#	.	.	.
3
4
5	#	.	.
6	#
7
8

萨丽总是垂直 (向上或者向下) 或水平 (向左或者向右) 地走。她可以从出发地 (总是记作 **A1**) 向下或者向右走。一旦萨丽选定了一个方向, 她就会一直走下去。如果她遇到棋盘边缘或者路障, 她就停下来, 并且转过 **90** 度。她不可能离开棋盘, 或者走进路障当中。并且, 萨丽从不跨过她已经经过的格子。当她再也不能走的时候, 她就停止散步。

这里是上面的棋盘上的一次散步路线图示：

	A	B	C	D	E	F	G	H
1	S	-----	+	#	.			
2	#		.	.
3
4	+	---	+
5	#	.	
6	#	
7	+	-----	+					
8	+	-----	+					

萨丽向右走，再向下，向右，向下，然后向左，再向上，最后向右走。这时她遇到了一个她已经走过的格子，她就停下来了。但是，如果她在 F5 格遇到路障后选择另外一条路——向我们看来是左边的方向转弯，情况就不一样了。

你的任务是计算并输出，如果萨丽聪明地选择她的路线的话，她所能够经过的最多格子数。

格式

PROGRAM NAME: snail

INPUT FORMAT

输入的第一行包括 N ——棋盘的大小，和 B ——路障的数量（ $1 \leq B \leq 200$ ）。接下来的 B 行包含着路障的位置信息。下面的样例输入对应着上面的示例棋盘。下面的输出文件表示问题的解答。注意，当 $N > 26$ 时，输入文件就不能表示 Z 列以后的路障了。

OUTPUT FORMAT

输出文件应该只由一行组成，即萨丽能够经过的最多格子数。

SAMPLE INPUT (file snail.in)

```
8 4
E2
A6
G1
F5
```

SAMPLE OUTPUT (file snail.out)

```
33
```

样例说明

```

      A B C D E F G H
1 S . . . . . # .
2 | . . . # . . .
3 | . . . +-----+
4 | . . . | . . |
5 +-----+ # . |
6 # . . . . . |
7 +----- |
8 +-----+
```

分析：
直接 DFS.本题目具有后效性。

Translate:USACO/fence3

Electric Fences 电网

Kolstad & Schrijvers

译 by Felicia Crazy

描述

农夫约翰已经决定建造电网。他已经把他的农田围成一些奇怪的形状，现在必须找出安放电源的最佳位置。

对于每段电网都必须从电源拉出一条电线。电线可以穿过其他电网或者跨过其他电线。电线能够以任意角度铺设，从电源连接到一段电网的任意一点上（也就是，这段电网的端点上或者在其之间的任意一点上）。这里所说的“一段电网”指的是呈一条线段状的电网，并不是连在一起的几段电网。若几段电网连在一起，那么也要分别给这些电网提供电力。

已知所有的 F ($1 \leq F \leq 150$) 段电网的位置（电网总是和坐标轴平行，并且端点的坐标总是整数， $0 \leq X, Y \leq 100$ ）。你的程序要计算连接电源和每段电网所需的电线的最小总长度，还有电源的最佳坐标。

电源的最佳坐标可能在农夫约翰的农田中的任何一个位置，并不一定是整数。

格式

PROGRAM NAME: fence3

INPUT FORMAT

第一行包括 F ——电网的数量。下面的 F 行每行包括两个 X, Y 对，表示这段电网的两个端点。

OUTPUT FORMAT

只有一行，输出三个浮点数，相邻两个之间留一个空格。假定你的电脑的输出库会正确地对小数进行四舍五入。

这三个数是：

电源最佳坐标的 X 值，电源最佳坐标的 Y 值，和需要的电线的总长度（要最小）。

SAMPLE INPUT (file fence3.in)

```
3
0 0 0 1
2 0 2 1
0 3 2 3
```

SAMPLE OUTPUT (file fence3.out)

```
1.0 1.6 3.7
```

问题分析：

模拟退火算法，可以加入随机化进行加速。基本思想很简单就是迭代逼近，USACO 官方给的解答是先随机找一个点然后在一个给定的步长范围内连续走，不断的更新这个点。从题目上看，这道题目的精度要求不高，那么就可以使用简单的退火算法来解决。首先是界定整张二维图的大小，然后用初始步长 STEP 网格搜索出当前的最好的点，那么可以知道，最优点一定在这个点的附近，接着 STEP 的大小缩小，如乘以 $1/10$ 等（可以用别的方式），我们在刚才找到的点的附近继续找，但是这回不是遍历整张图了，而是在刚才 STEP 所包围的最优点的范围内去找，这回的步长是 $STEP/10$ 。依次类推，这种退火算法就可以逐步找到最佳点。问题是控制在什么精度合适。首先题目要求输出一位小数，所以我们可以先把地图长度放大10倍，这样我们最后找到 $STEP = 1$ 的最佳点就是最优结果了。

不过，严格意义上说，退火算法应该与随机化结合，这道题目的数据范围不大，不加入随机化也可以很快 AC 的。这种分层次逐步逼近的思想在找到一个大规模地图上的较优解时，速度很快。

Translate:USACO/wissqu

Wisconsin Squares 威斯康星州的牧场

译 by Felicia Crazy

描述

威斯康星州的春天来了，是该把小奶牛们赶到小牧场上并把大奶牛们赶到北纬 40 度的大牧场上的时候了。

农夫约翰的牧场上有五种奶牛（括号内的是缩写）：格恩西奶牛（A），泽西奶牛（B），赫里福奶牛（C），黑安格斯奶牛（D），朗赫恩奶牛（E）。

这些奶牛群放养在一片 16 英亩的牧场上，每英亩上都有一小群奶牛，像下面这样排列成 4 x 4 的格子（用行和列标号）：

	1	2	3	4	
	+-----				
1		A	B	A	C
2		D	C	D	E
3		B	E	B	C
4		C	A	D	E

最初，牧场上的奶牛群总共有 3 群 A, 3 群 B, 4 群 C, 3 群 D, 3 群 E。
今年的 D 种小奶牛比去年多一群，C 种少一群，共有 3 群 A, 3 群 B, 3 群 C, 4 群 D, 3 群 E。

农夫约翰对于他牧场上的奶牛群的布局非常小心。这是因为如果同一种类型的奶牛群靠得太近，她们就乱来：她们聚集在栅栏边上抽烟，喝牛奶。如果她们在相同的格子上或者在临近的 8 个格子上，就是靠得太近了。

农夫约翰得用他的棕色旧福特皮卡把他的大奶牛群运出牧场，并把他的奶牛群运进牧场，皮卡一次只能运一群奶牛。他装上一群小奶牛，开车到牧场的一个方格中，卸下这群小奶牛，再装上这个格子上的那群大奶牛，开到

北纬 40 度的大牧场卸下来。他重复这样的操作 16 次，然后开车去杰克商店办理低脂酸奶的交易和家居装修。

帮助农夫约翰。他必须选择正确的顺序来替换他的奶牛群，使得他从不把一群小奶牛放入当前被同样类型奶牛占有的方格或者当前被同样类型奶牛占据的方格的临近方格。当然，一旦大奶牛走了，小奶牛就被安置好 (in place)，他必须小心以后的情况，要根据新的排列把奶牛群分开。

非常重要的提示：农夫约翰从过去的经验知道，他必须先移动 D 种奶牛群。

找出农夫约翰将他的小奶牛搬迁到她们的新牧场上的办法。输出 16 个连续的 奶牛群类型/行/列 的信息，使得这样的安排能够符合安全经验。

计算 4 x 4 牧场的最终排列总数和产生那些排列的方式的总数。

格式

TIME LIMIT: 5secs

PROGRAM NAME: wissqu

INPUT FORMAT

四行，每行四个字母，表示奶牛群。

OUTPUT FORMAT

16 行，每行分别由 奶牛群类型/行/列组成。如果有多解（一定有），那么你应该输出奶牛群类型按照字典序排列在最前面的那个解。如果不只一个解满足条件，那么你应该输出行信息按照字典需排列在最前面的那个解。如果仍然不只一个解满足条件，那么你应该输出列信息按照字典序排列在最前面的那个解。

最后多输出一行，包含能够由这个排列方式产生的排列的总数。

SAMPLE INPUT (file wissqu.in)

```
ABAC
DCDE
BEBC
CADE
```

[编辑] SAMPLE OUTPUT (file wissqu.out)

```
D 4 1
C 4 2
A 3 1
A 3 3
B 2 4
B 3 2
B 4 4
E 2 1
E 2 3
D 1 4
D 2 2
C 1 1
C 1 3
A 1 2
E 4 3
D 3 4
14925
```

问题分析：

没有加任何优化的暴力 DFS. 可以使用一种更快的方法，就是不判断周围8个方向的格子，而是用 $p[i][j][k]$ 表示 i, j 格子周围第 k 种牛的数目。