

第二十六届全国信息学奥林匹克竞赛

NOI 2009 第二试

竞赛时间：2009 年 7 月 29 日上午 8:00-13:00

题目名称	植物大战僵尸	管道取珠	描边
目录	pvz	ball	path
可执行文件名	pvz	ball	N/A
输入文件名	pvz.in	ball.in	path1.in~path10.in
输出文件名	pvz.out	ball.out	path1.out~path10.out
每个测试点时限	2 秒	2 秒	N/A
测试点数目	10	10	10
每个测试点分值	10	10	10
是否有部分分	无	无	有
题目类型	传统	传统	提交答案

提交源程序须加后缀

对于 Pascal 语言	pvz.pas	ball.pas	N/A
对于 C 语言	pvz.c	ball.c	N/A
对于 C++ 语言	pvz.cpp	ball.cpp	N/A

注意：最终测试时，所有编译命令均不打开任何优化开关

植物大战僵尸

【问题描述】

Plants vs. Zombies (PVZ) 是最近十分风靡的一款小游戏。Plants (植物) 和 Zombies (僵尸) 是游戏的主角, 其中 Plants 防守, 而 Zombies 进攻。该款游戏包含多种不同的挑战系列, 比如 Protect Your Brain、Bowling 等等。其中最为经典的, 莫过于玩家通过控制 Plants 来防守 Zombies 的进攻, 或者相反地由玩家通过控制 Zombies 对 Plants 发起进攻。

现在, 我们将要考虑的问题是游戏中 Zombies 对 Plants 的进攻, 请注意, **本题中规则与实际游戏有所不同**。游戏中有两种角色, Plants 和 Zombies, 每个 Plant 有一个攻击位置集合, 它可以对这些位置进行保护; 而 Zombie 进攻植物的方式是走到植物所在的位置上并将其吃掉。

游戏的地图可以抽象为一个 N 行 M 列的矩阵, 行从上到下用 0 到 $N-1$ 编号, 列从左到右用 0 到 $M-1$ 编号; 在地图的每个位置上都放有一个 Plant, 为简单起见, 我们把位于第 r 行第 c 列的植物记为 $P_{r,c}$ 。

Plants 分很多种, 有攻击类、防守类和经济类等等。为了简单的描述每个 Plant, 定义 *Score* 和 *Attack* 如下:

Score $[P_{r,c}]$ Zombie 击溃植物 $P_{r,c}$ 可获得的能源。若 *Score* $[P_{r,c}]$ 为非负整数, 则表示击溃植物 $P_{r,c}$ 可获得能源 *Score* $[P_{r,c}]$, 若为负数表示击溃 $P_{r,c}$ 需要付出能源 $-Score[P_{r,c}]$ 。

Attack $[P_{r,c}]$ 植物 $P_{r,c}$ 能够对 Zombie 进行攻击的位置集合。

Zombies 必须**从地图的右侧进入**, 且只能**沿着水平方向进行移动**。Zombies 攻击植物的唯一方式就是**走到该植物所在的位置**并将植物吃掉。因此 Zombies 的进攻总是从地图的右侧开始。也就是说, 对于第 r 行的进攻, Zombies 必须首先攻击 $P_{r,M-1}$; 若需要对 $P_{r,c}$ ($0 \leq c < M-1$) 攻击, 必须将 $P_{r,M-1}, P_{r,M-2} \dots P_{r,c+1}$ 先击溃, 并移动到位置 (r, c) 才可进行攻击。

在本题的设定中, Plants 的攻击力是无穷大的, 一旦 Zombie 进入某个 Plant 的攻击位置, 该 Zombie 会被瞬间消灭, **而该 Zombie 没有时间进行任何攻击操作**。因此, 即便 Zombie 进入了一个 Plant 所在的位置, 但该位置属于其他植物的攻击位置集合, 则 Zombie 会被瞬间消灭而所在位置的植物则安然无恙 (在我们的设定中, Plant 的攻击位置不包含自身所在位置, 否则你就不可能击溃它了)。

Zombies 的目标是对 Plants 的阵地发起进攻并获得最大的能源收入。每一次, 你可以选择一个可进攻的植物进行攻击。本题的目标为, 制定一套 Zombies 的进攻方案, 选择进攻哪些植物以及进攻的顺序, 从而获得最大的能源收入。

【输入文件】

输入文件 `pvz.in` 的第一行包含两个整数 N, M ，分别表示地图的行数和列数。

接下来 $N \times M$ 行描述每个位置上植物的信息。第 $r \times M + c + 1$ 行按照如下格式给出植物 $P_{r,c}$ 的信息：第一个整数为 $Score[P_{r,c}]$ ，第二个整数为集合 $Attack[P_{r,c}]$ 中的位置个数 w ，接下来 w 个位置信息 (r', c') ，表示 $P_{r,c}$ 可以攻击位置第 r' 行第 c' 列。

【输出文件】

输出文件 `pvz.out` 仅包含一个整数，表示可以获得的最大能源收入。注意，你也可以选择不进行任何攻击，这样能源收入为 0。

【输入样例】

```
3 2
10 0
20 0
-10 0
-5 1 0 0
100 1 2 1
100 0
```

【输出样例】

```
25
```

【样例说明】

在样例中，植物 $P_{1,1}$ 可以攻击位置 $(0,0)$ ， $P_{2,0}$ 可以攻击位置 $(2,1)$ 。

一个方案为，首先进攻 $P_{1,1}$ ， $P_{0,1}$ ，此时可以攻击 $P_{0,0}$ 。共得到能源收益为 $(-5)+20+10=25$ 。注意，位置 $(2,1)$ 被植物 $P_{2,0}$ 保护，所以无法攻击第 2 行中的任何植物。

【大致数据规模】

约 20% 的数据满足 $1 \leq N, M \leq 5$ ；

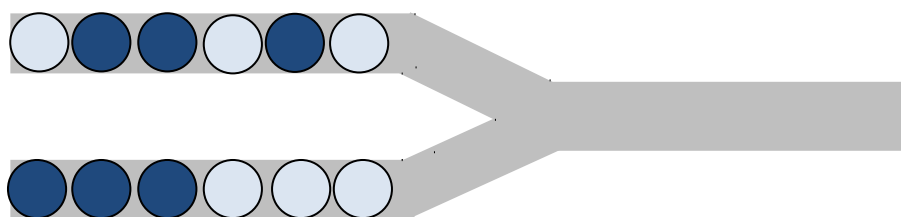
约 40% 的数据满足 $1 \leq N, M \leq 10$ ；

约 100% 的数据满足 $1 \leq N \leq 20, 1 \leq M \leq 30, -10000 \leq Score \leq 10000$

管道取珠

【问题描述】

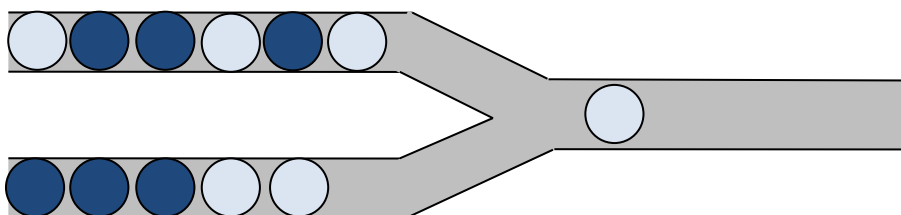
管道取珠是小 X 很喜欢的一款游戏。在本题中，我们将考虑该游戏的一个简单改版。游戏画面如图 1 所示：



(图 1)

游戏初始时，左侧上下两个管道分别有一定数量的小球（有深色球和浅色球两种类型），而右侧输出管道为空。每一次操作，可以从左侧选择一个管道，并将该管道中**最右侧的球**推入右边输出管道。

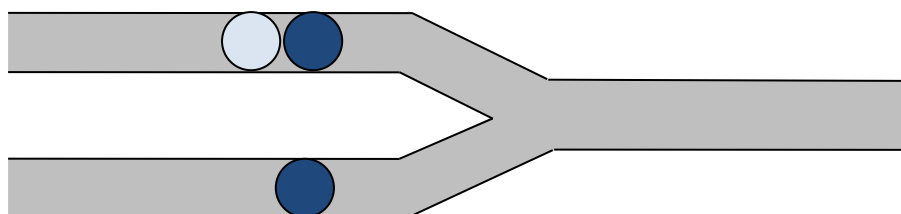
例如，我们首先从下管道中移一个球到输出管道中，将得到图 2 所示的情况。



(图 2)

假设上管道中有 n 个球，下管道中有 m 个球，则整个游戏过程需要进行 $n + m$ 次操作，即将所有左侧管道中的球移入输出管道。最终 $n + m$ 个球在输出管道中**从右到左形成输出序列**。

爱好数学的小 X 知道，他共有 $C(n+m, n)$ 种不同的操作方式，而不同的操作方式可能导致相同的**输出序列**。举个例子，对于图 3 所示的游戏情形：



(图 3)

我们用 A 表示浅色球，B 表示深色球。并设移动上管道右侧球的操作为 U，移动下管道右侧球的操作为 D，则共有 $C(2+1, 1)=3$ 种不同的操作方式，分别为 UUD, UDU, DUU；最终在输出管道中形成的**输出序列（从右到左）**分别为 BAB, BBA, BBA。可以发现后两种操作方式将得到同样的**输出序列**。

假设最终可能产生的不同种类的**输出序列**共有 K 种，其中第 i 种输出序列的产生方式(即不同的操作方式数目)有 a_i 个。聪明的小 X 早已知道，

$$\sum_{i=1}^k a_i = C(n+m, n)$$

因此，小 X 希望计算得到

$$\sum_{i=1}^k a_i^2$$

你能帮助他计算这个值么？由于这个值可能很大，因此只需要输出该值对 1024523 的取模即可(即除以 1024523 的余数)。

说明：文中 $C(n+m, n)$ 表示组合数。组合数 $C(a, b)$ 等价于在 a 个不同的物品中选取 b 个的选取方案数。

【输入文件】

输入文件 ball.in 第一行包含两个整数 n, m ，分别表示上下两个管道中球的数目。

第二行为一个 AB 字符串，长度为 n ，表示上管道中**从左到右**球的类型。其中 A 表示浅色球，B 表示深色球。

第三行为一个 AB 字符串，长度为 m ，表示下管道中的情形。

【输出文件】

输出文件 ball.out 仅包含一行，即为 $\sum_{i=1}^k a_i^2$ 除以 1024523 的余数。

【输入样例】

```
2 1
AB
B
```

【输出样例】

```
5
```

【样例说明】

样例即为文中(图 3)。共有两种不同的**输出序列**形式，序列 BAB 有 1 种产生方式，而序列 BBA 有 2 种产生方式，因此答案为 5。

【大致数据规模】

约 30% 的数据满足 $n, m \leq 12$ ；

约 100% 的数据满足 $n, m \leq 500$ 。

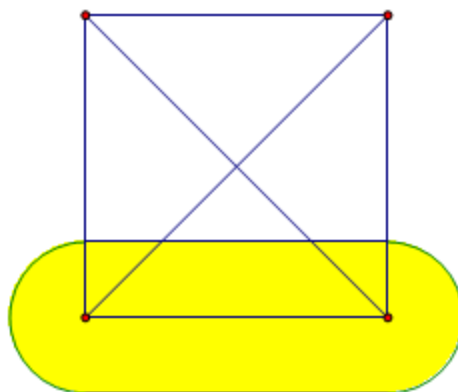
描边

【问题描述】

小 Z 自幼就酷爱数学。聪明的他特别喜欢研究一些数学小问题。

有一天，小 Z 在一张纸上选择了 n 个点，并用铅笔将它们两两连接起来，构成 $n(n-1)/2$ 条线段。由于铅笔很细，可以认为这些线段的宽度为 0。

望着这些线段，小 Z 陷入了冥想中。他认为这些线段中的一部分比较重要，需要进行强调。因此小 Z 拿出了毛笔，将它们重新进行了描边。毛笔画在纸上，会形成一个半径为 r 的圆。在对一条线段进行描边时，毛笔的中心(即圆心)将从线段的一个端点开始，沿着该线段描向另一个端点。下图即为在一张 4 个点的图中，对其中一条线段进行描边强调后的情况。



现在，小 Z 非常想知道在描边之后纸面上共有多大面积的区域被强调，你能帮助他解答这个问题么？

【输入文件】

这是一道提交答案型试题，所有的输入文件 path1.in~path10.in 已在相应目录下。

输入文件 path*.in 第一行包含一个正整数 n ，表示选择的点的数目。

第 2 至第 $n+1$ 行，第 $i+1$ 行有两个实数 x_i, y_i ，表示点 i 的坐标为 (x_i, y_i) 。

第 $n+2$ 行有一个正整数 m ，表示小 Z 认为比较重要的线段的条数。

第 $n+3$ 至第 $n+m+2$ 行，每行有两个正整数 a, b 表示一条线段。 a, b 两个数分别表示该线段的两个端点的编号。

第 $n+m+3$ 行，有一个实数 r ，表示毛笔在纸上形成的圆的半径。

第 $n+m+4$ 行，有四个实数 p_1, p_2, p_3, p_4 ，为评分使用的参数。

【输出文件】

输出文件 `path*.out` 仅包含一行，即为描边后被强调区域的总面积。

【输入样例】

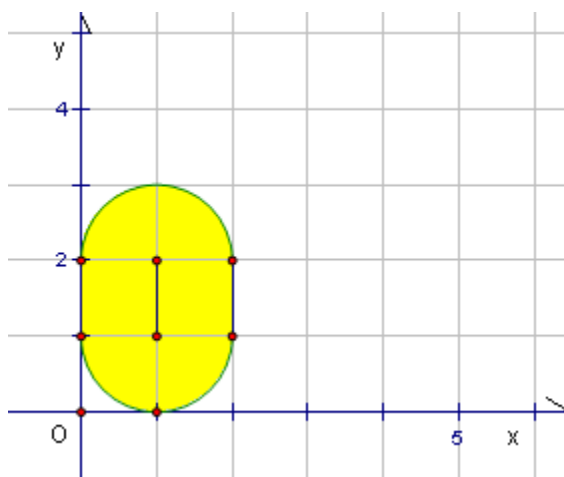
```
2
1 1
1 2
1
1 2
1
0.00001 0.001 0.1 1
```

【输出样例】

```
5.1415927
```

【样例说明】

如下图所示。



【评分标准】

每个测试点单独评分。

本题设有 4 个评分参数 p_1, p_2, p_3, p_4 ($p_1 < p_2 < p_3 < p_4$)，已在输入文件中给出。你的得分将按照如下规则给出：

若你的答案与标准答案相差不超过 p_1 ，则该测试点你将得到满分；

否则，若你的答案与标准答案相差不超过 p_2 ，则你将得到该测试点 70% 的分数；

否则，若你的答案与标准答案相差不超过 p_3 ，则你将得到该测试点 40% 的分数；

否则，若你的答案与标准答案相差不超过 p_4 ，则你将得到该测试点 10% 的分数；

否则该测试点你的得分为 0。