

棋局 (chess)

【题目背景】

在输了一晚上的麻将之后，小 z 和小 c 卸掉了手机上的所有牌类游戏。不过这怎么可能阻挡得了他们上课颓废的决心呢？现在他们的目光盯在了棋类游戏上，但他们两个除了天天下飞行器以外，几乎所有棋类游戏都只懂个大概规则。

“既然我们都会玩但只能玩一点点，不如我们自己搞个缝合怪出来吧！”

于是，在他们的精心脑洞之下，一个融合了围棋、象棋与军棋的奇妙游戏诞生了……

【题目描述】

游戏在一张长 n 行宽 m 列的网格形棋盘上进行，棋子落在网格的交叉点上。我们不妨记左上角的交叉点的坐标为 $(1, 1)$ ，右下角的交叉点坐标为 (n, m) 。

棋子分为黑白两色，对局双方各执一方棋子。

每个棋子除了颜色以外还有等级，不妨设 col_i 为棋子 i 的颜色， lv_i 为棋子 i 的等级。另外，棋盘上的网格线共有 4 种状态，对于第 i 条网格线，设其状态为 opt_i 。

轮到每方下棋时，他可以选择棋盘上的一个己方棋子沿网格线进行移动到另一个交叉点，称为走子。形式化定义走子的过程如下：选择一个坐标序列 $(x_0, y_0), (x_1, y_1), \dots, (x_k, y_k)$ ，其中 k 是任意选定的正整数， (x_0, y_0) 是棋子初始的位置， (x_k, y_k) 是棋子最终走到的位置，需要满足：

- 对于任意 $i = 0, 1, \dots, k - 1$ ，坐标 (x_i, y_i) 和 (x_{i+1}, y_{i+1}) 之间必须有网格线直接相连，也就是说走子必须沿着网格线走；
- 对于任意 $i \neq j$ ，必须有 $(x_i, y_i) \neq (x_j, y_j)$ ，也就是说走子过程中不能经过重复位置，特别地 $(x_0, y_0) \neq (x_k, y_k)$ ，也就是说不能原地不动（或走回原地）。
- 对于任意 $i = 1, \dots, k - 1$ ，坐标 (x_i, y_i) 上必须没有棋子，也就是说走子时不能越过已有的棋子。
- 若 (x_k, y_k) 上没有棋子，称为普通走子，否则称为吃子。在吃子过程中，设正在走的棋子颜色为 col_1 ，等级为 lv_1 ，被吃的棋子颜色为 col_2 ，等级为 lv_2 ，则必须满足 $col_1 \neq col_2$ ， $lv_1 \geq lv_2$ ，换句话说只能吃与自己颜色不同，且等级不高于自己等级的棋子。

需要注意的是，由上述定义可以得出，不允许棋子在吃子后继续向前走。

网格线的状态含义如下所述：

- 如果 $opt_i = 0$ ，代表此路不通，走子时不能经过这条网格线。
- 如果 $opt_i = 1$ ，代表这条网格线是一条“普通道路”，每次走子时棋子最多只能经过 1 条普通道路。
- 如果 $opt_i = 2$ ，代表这条网格线是一条“直行道路”，每次走子时棋子可以经过任意条直行道路，但只能一直沿横向或一直沿纵向走，不能转弯。如沿直行道路从 $(1, 1)$ 经过 $(1, 2)$ 走到 $(1, 3)$ 是可以的，但是从 $(1, 1)$ 经过 $(1, 2)$ 走到 $(2, 2)$ 不行。

- 如果 $opt_i = 3$ ，代表这条网格线是一条“互通道路”，每次走子时棋子可以经过任意条互通道路，且中途可任意转弯。

同时规定在一次走子过程中，**棋子经过的网格线的状态必须全部相同**，比如从 $(1, 1)$ 经过直行道路走到 $(1, 2)$ 再经过互通道路走到 $(1, 3)$ 是不允许的。

至于如何判断胜负等其他细节，与本题无关，故略去。

小 z 和小 c 开发出这款棋类游戏后，为了提升水平，想了一个训练的策略：一开始棋盘是空的，然后小 c 会每次往棋盘的某个空交叉点上放一枚棋子，小 z 需要快速计算出：若选择这枚新放上的棋子进行一次走子，棋盘上一共有多少个位置是能被走到的？注意，因为这只是思维训练，他们并不会真的走这枚棋子。

可怜的小 z 发现他的计算力不足以算出这个问题，只好向你求助。

【输入格式】

从文件 `chess.in` 中读入数据。

每个测试点由多组数据组成。

第一行：一个正整数 T 表示数据组数。

对于每组数据：

第一行：3 个正整数 n, m, q ，分别表示棋盘的行数、列数和游戏的轮数。

接下来 n 行，每行为一个长 $m - 1$ 的字符串，每个字符为 $0, 1, 2, 3$ 中的一个，第 i 行第 j 个字符表示交叉点 (i, j) 连向交叉点 $(i, j + 1)$ 的网格线状态。

接下来 $n - 1$ 行，每行为一个长 m 的字符串，每个字符为 $0, 1, 2, 3$ 中的一个，第 i 行第 j 个字符表示交叉点 (i, j) 连向交叉点 $(i + 1, j)$ 的网格线状态。

接下来 q 行，每行 4 个非负整数 col_i, lv_i, x_i, y_i ，表示在第 i 轮有一枚颜色为 col_i ，等级为 lv_i 的棋子放在了交叉点 (x_i, y_i) 上。其中 $col_i = 0$ 表示黑子， $col_i = 1$ 表示白子。保证之前交叉点 (x_i, y_i) 上没有棋子。

【输出格式】

输出到文件 `chess.out` 中。

对于每组数据输出 q 行，每行一个非负整数，表示第 i 枚棋子放置后能走到的交叉点数量。

【样例 1 输入】

```

1 1
2 3 3 5
3 13
4 22
5 23

```

```
6 010
7 233
8 0 1 2 3
9 1 2 2 1
10 1 3 1 2
11 0 2 3 2
12 1 3 2 2
```

【样例 1 输出】

```
1 4
2 3
3 3
4 3
5 2
```

【样例 1 解释】

放置棋子 1 后，它能走到的位置为 $(2, 1), (2, 2), (3, 2), (3, 3)$ 。

放置棋子 2 后，它能走到的位置为 $(2, 2), (2, 3), (3, 1)$ 。

放置棋子 3 后，它能走到的位置为 $(1, 1), (1, 3), (2, 2)$ 。

放置棋子 4 后，它能走到的位置为 $(2, 2), (3, 1), (3, 3)$ 。

放置棋子 5 后，它能走到的位置为 $(2, 3), (3, 2)$ 。

【样例 2 输入】

```
1 2
2 2 3 4
3 22
4 33
5 123
6 0 2 1 2
7 0 1 2 1
8 1 2 1 3
9 0 3 2 2
10 3 2 3
11 3
```

```

12 1
13 3
14 32
15 32
16 0 2 1 2
17 1 2 3 2
18 0 1 2 2

```

【样例 2 输出】

```

1 3
2 4
3 4
4 2
5 5
6 5
7 1

```

【样例 3】

见选手目录下的 *chess/chess3.in* 与 *chess/chess3.ans*。

【样例 4】

见选手目录下的 *chess/chess4.in* 与 *chess/chess4.ans*。

【数据范围】

测试点编号	$n \times m \leq$	$q \leq$	特殊性质
1 ~ 2	100	50	无
3 ~ 6	5000	2000	
7 ~ 8	2×10^5	10^5	不存在“直行道路”与“互通道路”
9 ~ 11			不存在“互通道路”
12 ~ 14			不存在“直行道路”
15 ~ 16			$lv_i = i$
17 ~ 18			$lv_i = q - i + 1$
19 ~ 21		2000	$n, m \leq 1000$
22 ~ 25		10^5	无

对于 100% 的数据, $1 \leq T \leq 5$, $2 \leq n, m \leq 10^5$, $4 \leq n \times m \leq 2 \times 10^5$, $1 \leq q \leq \min\{10^5, n \times m\}$, $1 \leq lv_i \leq q$, $1 \leq x_i \leq n$, $1 \leq y_i \leq m$, $col_i \in \{0, 1\}$ 。

注: 由于本题输入输出规模较大, 建议使用较为快速的输入输出方式。