



HPC<sup>3</sup> 2024

## 问题一，中文

### 狩猎之王

最高分：70

---

你的父亲是一位著名的棋子猎人，最近去世了，他没有完成毕生的使命——清除大陆上所有的野生棋子。你只希望用你父亲的地图、他的“棋子箱”和他的棋子捕捉装置来完成他的使命。

网格( $1 \leq N \leq 10^4$ ,  $1 \leq M \leq 10^4$ )上寻找棋子  $N \times M$ 。棋子位于网格上的一个点上。

狩猎将按如下方式进行：

- 第一块棋子将被放置在网格上的随机点
- 直到你抓住它为止，会发生搜寻轮次：你将一个装置放置在网格上的某个点，然后该棋子就会移动。
- 一旦抓住它，所有装置都将被移除，并且下一个装置将随机放置在网格上。

共有 5 种类型的棋子，每种棋子都以独特的模式沿网格移动：

- 棋子移动到基本相邻的点。
- 国王移动到基数或对角线相邻的点。
- 主教沿单一方向对角线移动任意数量的点。
- 车垂直于其行或列中的任意点移动。
- 皇后可以垂直移动到其行或列中的任意点，或者沿单一方向对角线移动到任意数量的点。

您可以使用两种设备：陷阱和传感器。陷阱会捕捉任何穿过或在其上移动的棋子，但棋子只有在必要时才会这样做。只要棋子穿过或在其上移动，或者在其基数或对角线上相邻的方格移动，传感器就会提醒您。

除了避开陷阱之外，棋子是简单的生物，它们必须在每个狩猎回合中从可用点中随机选择一个点并移动到该点。

您的食物有限，因此必须在 500,000 次狩猎中捕获所有食物。

## 笔记

- 这个问题是交互式的，对于每一个测试用例，你的程序将重复地给出输出并接收依赖于过去输出的输入。
- 这个问题具有随机性。因此，它可能会出现不确定的结果。但是，正确的解决方案始终是最优的，因此始终会获得最高分。

## 子问题 1

你父亲的“棋子表”包含一个有序列表，其中列出了你将要捕猎的所有棋子，该列表用一定 $S$ 长度的字符串表示 $p$  ( $1 \leq p \leq 50$ )，其中字符串中的每个字符 $c$ 代表一个棋子以及它何时到达（“P”代表兵，“K”代表王，“B”代表象，“R”代表车，“Q”代表后）。

给定 $N$ 、 $M$ 和 $S$ 。进行搜寻轮次以在 500,000 个搜寻轮次内捕获所有棋子。

### 第一种输入格式

每个输入的第一行包含 3 个整数 $N$ 、 $M$ 和 $p$ 。

每个输入的第二行包含一个长度为  $p$  的字符串 $S$ 。

```
N M p
S
```

### 狩猎回合输出格式

每个输出的第一行包含 1 个二进制值 $D$ 。每个输出的第二行包含 2 个整数： $x$  和 $y$ 。

```
D
x y
```

$D$  是您要放置的设备（0 表示陷阱，1 表示传感器）  $x, y$  是设备的位置。如果那里已经有设备，它将被替换。

## 狩猎回合输入格式

每个输入的第一行包含2个整数 $R$  和 $k$ 。每个输入的第二行包含 $k$  整数对：数组的内容 $P$ 。

```
R k
P[0][0] P[0][1] P[1][0] P[1][1] ... P[k-1][0] P[k-1][1]
```

$R$  是棋子移动的结果，介于  $-1$  和  $3$  之间。

- 如果为  $-1$ ，则表示您已超出 500,000 轮限制并且已经超出时间。
- 如果为  $0$ ，则棋子移动，不发生其他任何事情。
- 如果为  $1$ ，则表示该棋子触发了一些传感器。这意味着 中的每个对都是 $P$ 棋子经过或到达的传感器范围内的一个点。
- 如果是  $2$ ，则表示该棋子已被吃掉。要么是棋子被迫落入陷阱，要么是最后放置的装置是陷阱。所有装置均被移除。
- 如果是  $3$ ，则该棋子已被吃掉并且是最后一枚棋子，这意味着你赢了！

## 示例测试用例

输入 1

```
3 3 2
PK
```

输出 H1 : R1

```
1
2 2
```

输入 H1 : R1

```
1 1
1 2
```

输出 H1 : R2

```
0
1 2
```

输入 H1 : R2

```
2 0
```

输出H2 : R1

```
1
2 2
```

输入 H2 : R1

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

输出 H2 : R2

```
0
3 1
```

输入 H2 : R2

```
3 0
```

您在 3x3 网格上先追击一个兵，然后追击一个车。在所有情况下，最佳做法是在网格中间放置一个传感器，因为这会告诉您棋子移动后的准确位置，让您可以将其困住。

## 子问题 2

清除完家里的所有棋子后，您就转移到了其他土地。这里的棋子已经进化出了电念力，这导致它们与子问题 1 中的棋子有 3 个不同之处：首先，它们太危险了，无法接近，因此您无法在它们当前的位置放置陷阱。其次，它们可以远程停用传感器，因此您无法放置任何传感器。最后，它们周围有明亮的光环，这意味着您始终知道它们的位置。

你父亲的“零钱”和子问题 1 中的一样，的最大值 $N, M$ 是 100.

给定 $N, M$ 和 $S$ 。进行狩猎轮次以在5,000,000个狩猎轮次内捕获所有棋子。

### 第一个输入格式

每个输入的第一行包含 5 个整数 $N, M, p, x$ 和 $y$ 。

每个输入的第二行包含一个长度为  $p$  的字符串  $S$ 。

```
N M p x y
S
```

这件作品的起点在哪里 $x, y$

### 狩猎回合输出格式

每个输出的第一行也是唯一一行包含 2 个整数： $x$  和  $y$ 。

```
x y
```

$x$ ,  $y$  是您将放置陷阱的位置。

## 狩猎回合输入格式

每个输入的第一行包含 1 个整数  $R$ 。每个输入的第二行包含 2 个整数： $x$  和  $y$ 。

```
R  
x y
```

$R$  是棋子移动的结果，介于  $-1$  和  $2$  之间。

- 如果为  $-1$ ，则表示您已超出 5,000,000 轮限制并且已经超出时间。
- 如果为  $0$ ，棋子移至  $x$ ,  $y$ 。
- 如果为  $1$ ，表示该棋子已被吃掉。所有陷阱均被移除，并且  $x$  是  $y$  下一个棋子的起始位置。
- 如果是  $2$ ，则该棋子已被吃掉并且是最后一枚棋子，这意味着你赢了！

## 示例测试用例

### 输入 1

```
3 3 2 1 1  
PK
```

### 输出 H1 : R1

```
2 2
```

### 输入H1 : R1

0  
2 1

输出 H1 : R2

1 1

输入 H1 : R2

0  
3 1

输出 H1 : R3

3 2

输入 H1 : R3

0  
2 1

输出 H1 : R4

3 1

输入 H1 : R4

1  
2 2

输出 H2 : R1

1 2

输入 H2 : R1

输出H2 : R2

输入 H2 : R2

输出 H2 : R3

输入 H2 : R3

输出 H2 : R3

输入 H2 : R3

[illegible]

请注意，尽管王是机动性更强的棋子，但它被抓住的速度更快，因为它随机移动，导致自己进入了更糟糕的位置。