

# CSP-S 2019 模拟赛

## 第二试

时间：2019 年 11 月 10 日 08:00 ~ 11:30

AHSOFNU

luhong & hotwords & PinkRabbit

试题名称	扫雷	五彩树	大洗牌
输入文件名	minesweeper.in	colorful.in	shuffle.in
输出文件名	minesweeper.out	colorful.out	shuffle.out
每个测试点时限	1.0 秒	3.0 秒	1.0 秒
内存限制	512 MB	512 MB	512 MB
测试点数目	20	20	20
测试点是否等分	是	是	是

提交源程序文件名

对于 C++ 语言	minesweeper.cpp	colorful.cpp	shuffle.cpp
-----------	-----------------	--------------	-------------

### 注意事项

1. 评测在 64 位 Windows 10 下的 lemon 进行，配置为 CPU Intel(R) Core(TM) i5-8300H @ 2.30 GHZ, RAM 8 GB。
2. 只能使用 C++ 语言提交，不开启 C++11，不开启 O2 优化。
3. 评测时 g++ 版本为 4.9.2；栈空间与运行内存上限相同。
4. 提交时请各校将代码包发送至邮箱 693511570@qq.com。
5. 结果比较方式为全文比较，忽略行末空格和文末空行。

## 扫雷 (minesweeper)

### 【题目描述】



你需要编写一个扫雷交互器，获取地图信息，依次读入玩家操作并返回对应结果。

扫雷的局面是一个  $n \times m$  的矩形，其中一些位置为地雷而另一些位置为空地，扫雷局面将以字符矩阵的形式输入。将第  $i$  行第  $j$  列的位置记作  $\langle i, j \rangle$ 。特别地，令  $k$  为地雷的数量，保证有  $0 < k < n \times m$ 。

一开始玩家无法得知除了  $n, m, k$  之外的扫雷局面的任何信息。

你需要维护名为**玩家地图**的字符矩阵，初始时矩阵中所有元素为 `_`（下划线）。玩家将进行  $q$  次操作，每次将选取一个位置  $\langle x, y \rangle$ ，并用以下三种方式之一点击（若**游戏结束**，你应该忽略游戏结束后的所有玩家操作，即判定这些操作为**无效操作**，反之即为**有效操作**）：

#### 1. 左键点击：

若  $\langle x, y \rangle$  已经被**打开**或**玩家地图**中这个位置为 `P`（`P` 表示旗子），则不进行任何操作；否则若  $\langle x, y \rangle$  为地雷，则**游戏失败**；否则对  $\langle x, y \rangle$  进行**打开操作**。

## 2. 右键点击:

若  $\langle x, y \rangle$  已经被打开, 则不进行任何操作; 否则

若玩家地图中位置  $\langle x, y \rangle$  为  $\_$ , 将其改为 P,

若玩家地图中位置  $\langle x, y \rangle$  为 P, 将其改为 ?,

若玩家地图中位置  $\langle x, y \rangle$  为 ?, 将其改为  $\_$ 。

## 3. 中键点击:

若  $\langle x, y \rangle$  未被打开, 或玩家地图中这个位置周围相邻的 8 个位置的 P 的个数不等于玩家地图中该位置的数值, 则不进行任何操作; 否则对于  $\langle x, y \rangle$  周围相邻 8 个未被打开且在玩家地图中不是 P 的位置, 如果存在至少一个位置是地雷, 则游戏失败; 否则对这些位置进行打开操作。

## 打开操作:

对位置  $\langle x, y \rangle$  进行的打开操作按照如下方式进行:

1. 将  $\langle x, y \rangle$  标记为被打开。
2. 在玩家地图中位置  $\langle x, y \rangle$  改为  $c$  ( $0 \leq c \leq 8$ ), 表示这个位置周围相邻的 8 个位置的地雷数量。
3. 如果  $c = 0$ , 则对其周围相邻的 8 个未被打开且在玩家地图中不是 P 的位置进行打开操作。
4. 这个操作是递归进行的, 直到所有子操作都结束后, 本次打开操作才算结束。

## 游戏结束, 游戏结束有以下三种情况:

1. 游戏失败: 即上述规则中的情况, 试图对为地雷的位置进行打开操作就会导致游戏失败。
2. 游戏胜利: 若某一次操作结束后, 未被打开的位置个数恰好为  $k$ , 则此次操作后游戏胜利。

3. **退出游戏**：若玩家操作结束，但上述两种情况均未出现，则视作玩家退出游戏。

在每一次操作后，你需要返回结果，具体规则如下：

- 若此次操作为**无效操作**，返回 INVALID；否则
- 若此次操作后**游戏失败**，返回 LOSE；否则
- 先返回 RUNNING:，然后在同一行返回用中括号包含的，**玩家地图**中有更改的位置以及更改后的值，格式为  $\langle x, y, val \rangle$ ；更改的位置按照  $x$  坐标为第一关键字从小到大， $y$  坐标为第二关键字从小到大的顺序排序，相邻两个更改的位置用 `,` 隔开。例如某一次操作后可能返回：RUNNING: [ $\langle 2, 2, 2 \rangle, \langle 2, 3, 1 \rangle, \langle 3, 2, 1 \rangle, \langle 3, 3, 0 \rangle$ ]；或 RUNNING: []；或 RUNNING: [ $\langle 3, 3, ? \rangle$ ]。注意其中空格的位置。
- 若此次操作后**游戏胜利**，再在新的一行返回 WIN。
- 若此次操作后**退出游戏**，再在新的一行返回 QUIT。

### 【输入格式】

从文件 *minesweeper.in* 中读入数据。

**本题输入文件包含多组数据。**

第一行一个正整数  $T$  表示数据组数，接下来每  $n + q + 2$  行（意义见下）表示一组数据。

每组数据第一行两个正整数  $n, m$  分别表示扫雷局面的高度和宽度。

每组数据接下来  $n$  行，第  $i$  行一个长度为  $m$  的字符串，仅包含 `_` 和 `*` 两种字符。如果第  $j$  个字符为 `*`，则表示第  $i$  行第  $j$  列为地雷，否则为空地。

每组数据接下来若干行，每行三个正整数  $op, x, y$  表示玩家的一次操作，具体操作见题目描述。

每组数据最后一行一个数 0，表示玩家操作结束。令玩家操作次数为  $q$ 。

**【输出格式】**

输出到文件 *minesweeper.out* 中。

对于每组数据，输出若干行，每行表示一次操作的返回结果，若在某一次操作后**游戏结束**，请输出对应的结果。

相邻的两组数据之间使用一行 ===== (10 个 = 字符，不包含引号) 隔开。

**【样例 1 输入输出】**

见选手目录下的 *minesweeper/minesweeper1.in* 与  
*minesweeper/minesweeper1.ans*。

**【样例 1 解释】**

以下是对样例 1 的五组数据的解释，每组数据从左到右依次描述了扫雷局面以及每一次**有效操作**后的玩家矩阵。

第一组数据：

```
__* | ___ | 01_
___ | ___ | 011
___ | ___ | 000
```

第二组数据：

```
_*_ | ___ | ___ | __2 | _P2 | _P2 | 1P2
__* | ___ | 12_ | 12_ | 12_ | 12P | 12P
___ | ___ | 01_ | 01_ | 01_ | 01_ | 011
```

第三组数据：

```
*** | ___ | ***
___ | ___ | _6_
*** | ___ | ***
```

第四组数据（玩家矩阵中不包含 x 字符，最后一个矩阵仅作参考用，不是真实的玩家矩阵）：

```

_ _ _ | _ _ _ | _ _ _
_ * _ | _ _ _ | _ x _
_ _ _ | _ _ _ | _ _ _

```

第五组数据（玩家矩阵中不包含 x 字符，最后一个矩阵仅作参考用，不是真实的玩家矩阵）：

```

_ _ _ | _ _ _ | _ _ _ | _ P _ | 0P1
_ _ * | _ _ _ | _ 1 _ | _ 1 _ | 01x
_ _ _ | _ _ _ | _ _ _ | _ _ _ | 011

```

## 【样例 2】

见选手目录下的 *minesweeper/minesweeper2.in* 与 *minesweeper/minesweeper2.ans*。

## 【数据范围与提示】

对于所有测试点： $1 \leq T \leq 30$ ， $3 \leq n, m \leq 200$ ， $0 \leq q \leq 10000$ ， $op \in \{1, 2, 3\}$ ， $1 \leq x \leq n$ ， $1 \leq y \leq m$ ，  
保证输入的字符串仅包含 \_ 和 \* 两种字符。

测试点编号	$n$	$m$	$k$	特殊限制
1	= 9	= 9	= 10	使用真实游戏生成
2	= 16	= 16	= 40	使用真实游戏生成
3	= 16	= 30	= 99	使用真实游戏生成
4	≤ 40	≤ 40	≤ 320	使用真实游戏生成
5 ~ 8	≤ 200	≤ 200	≤ 10000	op = 1
9 ~ 12	≤ 200	≤ 200	≤ 10000	op ∈ {1, 2}
13 ~ 16	≤ 200	≤ 200	≤ 10000	op ∈ {1, 3}
17 ~ 20	≤ 200	≤ 200	≤ 10000	无

## 五彩树 (colorful)

### 【题目描述】

给定一棵  $n$  个节点的树，节点编号为  $1 \sim n$ 。

每个节点都染上了一种颜色，总共有  $m$  种不同的颜色，编号为  $1 \sim m$ 。  
记节点  $i$  的颜色为  $c_i$ 。

小 X 喜欢颜色，他想要选出这棵树的一个连通的部分  $S$ ，并且  $S$  中的节点必须包含至少  $k$  种不同的颜色。即  $S$  必须满足其是原树的一个连通的导出子图，并且集合  $\{c_u \mid u \in S\}$  的大小至少为  $k$ 。

但是小 Y 讨厌颜色，她让小 X 把除了  $S$  中的节点之外的所有节点的颜色都擦除，并且呆在一个节点  $u$ ，她想要最大化节点  $u$  与任意的有颜色的节点之间的距离的最小值。即最大化  $\min_{x \in S} \text{dis}(u, x)$ ， $\text{dis}(x, y)$  表示节点  $x$  与节点  $y$  之间的距离，这里距离定义为最少经过的边数。

小 X 还没有确定选择  $S$  的方案，他求助于你，你需要满足上述所有条件，并且最大化  $u$  与  $S$  中的节点的距离。

### 【输入格式】

从文件 *colorful.in* 中读入数据。

第一行三个正整数  $n, m, k$ ，意义见题目描述。

第二行  $n$  个正整数  $c_{1 \dots n}$  依次表示每个节点的颜色。

接下来  $n - 1$  行，每行两个正整数  $x, y$  表示一条连接节点  $x$  与节点  $y$  的边。

### 【输出格式】

输出到文件 *colorful.out* 中。

输出一行一个数表示最大距离。

【样例输入】

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

【样例输出】

3

【数据范围与提示】

对于所有测试点： $1 \leq k \leq m \leq n \leq 10^6$ ， $1 \leq c_i \leq m$ 。  
保证  $1 \leq x, y \leq n$ ， $x \neq y$  且给出的连边情况形成一棵树。  
每个测试点的具体限制见下表：

测试点编号	$n \leq$	$m \leq$	特殊限制
1 ~ 4	200	200	无
5 ~ 6	$10^6$	1	无
7 ~ 10	$10^6$	50	无
11 ~ 12	$10^6$	$10^6$	$m = n$
13 ~ 14	$10^6$	$10^6$	$k = m$
15 ~ 20	$10^6$	$10^6$	无



## 大洗牌 (shuffle)

### 【题目描述】

有  $n$  张纸牌，简易起见将它们编号为  $1 \sim n$ 。

一开始所有的牌按顺序排列，从上往下数第  $i$  张牌的编号为  $i$ 。

魔术师 Trixie 会施展一种神秘的洗牌魔术，施展一次魔术会让原来从上往下数的第  $i$  张纸牌在新的牌堆中排在从上往下数第  $p_i$  个位置，保证  $p_i$  是  $1$  到  $n$  的排列。

现在 Trixie 按顺序施展了这个魔术  $k$  次，使得牌堆中从上往下数第  $i$  张牌的编号为  $a_i$ 。

现在你只得到了  $a_i$  的值，但是不知道  $p_i$  的值，Trixie 打赌你不可能还原出  $p_i$  的值。

如果你猜对了  $p_i$  的值，Trixie 就答应用魔术让你的得分增加 100 分，但是如果你猜错了……

你可不想接受这么大的惩罚，所以你还想要算出有多少个  $p_i$  满足条件，以权衡得失。注意：你也可以不计算这个值，但只能获得该测试点 70% 的分数。

Trixie 的魔术万无一失，但是她可不一定诚实，所以要是没有满足条件的  $p_i$  你就可以揭穿她的把戏；否则你还需要输出一组满足条件的  $p_i$ 。

### 【输入格式】

从文件 `shuffle.in` 中读入数据。

第一行两个正整数  $n, k$ ，意义见题目描述。

第二行  $n$  个正整数  $a_{1 \dots n}$  依次表示最终牌堆中牌的编号。

### 【输出格式】

输出到文件 `shuffle.out` 中。

你可以选择计算方案数或仅输出方案，如果你选择了仅输出方案，请在第一行输出 0，否则输出 1。

如果你选择了仅输出方案，若存在至少一组满足条件的  $p_i$ ，在第二行输出 1，否则输出 0；否则，若存在至少一组满足条件的  $p_i$ ，在第二行输出一个数表示满足条件的  $p_i$  的个数，对 998244353 取模，否则输出 -1。

如果存在至少一组满足条件的  $p_i$ ，在第三行输出  $n$  个数，表示一组满足条件的  $p_i$ 。否则输出  $n$  个 0。

若有多组解，输出任意一组均可。

注意：无论你选择了计算方案数或是仅输出方案，如果格式与上文所述的格式不符或输出的方案不合法，都无法获得任何分数。

### 【样例 1 输入】

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

### 【样例 1 输出】

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

### 【样例 2 输入】

```
6 2
1 2 3 4 5 6
```

### 【样例 2 输出】

```
1
76
6 5 4 3 2 1
```

【样例 3 输入】

2 2  
2 1

【样例 3 输出】

1  
-1  
0 0

【数据范围与提示】

对于所有测试点： $1 \leq n \leq 10^5$ ， $2 \leq k \leq 10^{18}$ 。  
保证  $a_i$  是 1 到  $n$  的排列。  
每个测试点的具体限制见下表：

测试点编号	$n$	$k$	特殊限制
1 ~ 3	$\leq 10$	$\leq 10$	无
4 ~ 5	$\leq 10$	$\leq 10^{18}$	无
6	$\leq 10^5$	$= 2$	无
7 ~ 8	$\leq 10^5$	$= 3$	无
9 ~ 12	$\leq 10^5$	$\leq 10^{18}$	$k$ 是质数
13 ~ 15	$\leq 10^5$	$\leq 10^{18}$	$a_i = i \bmod n + 1$
16 ~ 20	$\leq 10^5$	$\leq 10^{18}$	无