

# 真 • NOIP2019 模拟赛

lizehon

2019 年 8 月 7 日

题目名称	贪围蛇	进攻	填数游戏
题目类型	传统型	传统型	传统型
文件名称	snake.cpp	attack.cpp	game.cpp
输入文件	snake.in	attack.in	game.in
输出文件	snake.out	attack.out	game.out
测试点数量	10	10	20
时间限制	1s	2s	1s
空间限制	512MB	512MB	512MB
下发样例	有	有	无
编译选项	-lm -O2		

## 注意事项

1. 题目难度与顺序无关
2. 发现原题不要大喊”这不是 xxx 的原题吗”，可以闷声发大财
3. AK 后可以提前离场, 不要大声喧哗
4. Best wishes !

# 1 贪围蛇 snake

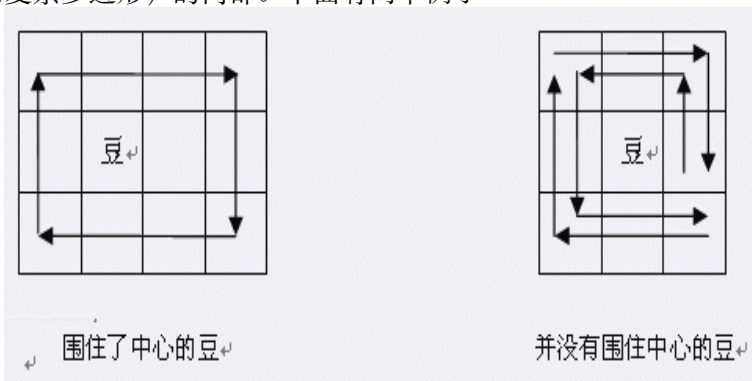
## 1.1 题目背景

本来有一个很搞笑的故事 (其实是三个), 可惜后来被我删了.

## 1.2 题目描述

在一个  $N \times M$  的矩阵方格内分布着  $D$  颗豆子, 每颗豆有不同的分值  $V_i$ 。游戏者可以选择任意一个方格作为起始格, 每次移动可以随意的走到相邻的四个格子, 直到最终又回到起始格。最终游戏者的得分为所有被路径围住的豆的分值总和减去游戏者移动的步数。矩阵中某些格子内设有障碍物, 任何时刻游戏者不能进入包含障碍物或豆子的格子。游戏者可能的最低得分为 0, 即什么都不做。

注意路径包围的概念, 即某一颗豆在路径所形成的多边形 (可能是含自交的复杂多边形) 的内部。下面有两个例子:



第一个例子中, 豆在路径围成的矩形内部, 所以豆被围住了。第二个例子中, 虽然路径经过了豆的周围的 8 个格子, 但是路径形成的多边形内部并不包含豆, 所以没有围住豆子。

注意每个格子可以经过多次

## 1.3 输入格式

第一行两个整数  $N$  和  $M$ , 为矩阵的边长。第二行一个整数  $D$ , 为豆子的总个数。

第三行包含  $D$  个整数  $V_1$  到  $V_D$ , 分别为每颗豆子的分值。

接着  $N$  行有一个  $N \times M$  的字符矩阵来描述游戏矩阵状态，0 表示空格，# 表示障碍物。而数字 1 到 9 分别表示对应编号的豆子。

## 1.4 输出格式

仅包含一个整数，为最高可能获得的分值。

## 1.5 样例

### 1.5.1 snake1.in

```
3 8
3
30 -100 30
00000000
010203#0
00000000
```

### 1.5.2 snake1.out

```
38
```

### 1.5.3 snake2.in/snake2.out

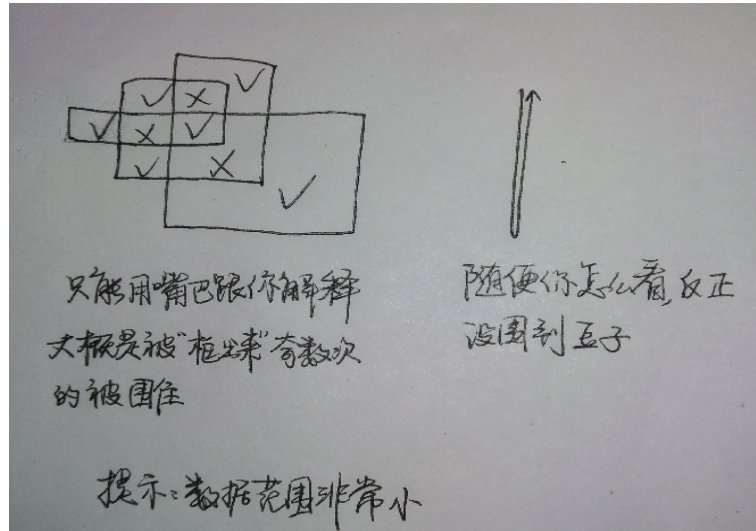
见下发样例

## 1.6 数据范围

100% 的数据满足  $1 \leq D \leq 9; 1 \leq N, M \leq 10; -10000 \leq V_i \leq 10000$ 。

数据有梯度。(人话: 有很水的数据)

## 1.7 补充说明



## 2 地球危机 attack

### 【题目描述】

三体人将对地球发起攻击。为了抵御攻击，地球人派出了  $A \times B \times C$  艘战舰，在太空中排成一个  $A$  层  $B$  行  $C$  列的立方体。其中，第  $i$  层第  $j$  行第  $k$  列的战舰（记为战舰  $(i, j, k)$ ）的生命值为  $d[i][j][k]$ 。三体人将会对地球发起  $m$  轮“立方体攻击”，每次攻击会对一个小立方体中的所有战舰都造成相同的伤害。

具体地，第  $t$  轮攻击用 7 个参数  $la_t, ra_t, lb_t, rb_t, lc_t, rc_t, h_t$  描述：所有满足  $i \in [la_t, ra_t], j \in [lb_t, rb_t], k \in [lc_t, rc_t]$  的战舰  $(i, j, k)$  会受到  $h_t$  的伤害。如果一个战舰累计受到的总伤害超过其生命值，那么这个战舰会爆炸。地球指挥官希望你能告诉他，第一艘爆炸的战舰是在哪一轮攻击后爆炸的。

### 【输入格式】

第一行包括 4 个正整数  $A, B, C, m$ ；

第二行包含  $A \times B \times C$  个整数，其中第  $((i-1) \times B + (j-1)) \times C + (k-1) + 1$  个数为  $d[i][j][k]$ ；

人性化地说，你可以这样写：

```
rep(i,1,A) rep(j,1,B) rep(k,1,C) d[i][j][k] = read();
```

接下来  $m$  行，每行包含 7 个正整数  $la_t, ra_t, lb_t, rb_t, lc_t, rc_t, h_t$ 。

### 【输出格式】

输出第一个爆炸的战舰是在哪一轮攻击后爆炸的。保证一定存在这样的战舰。

### 【样例 1 输入】

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

### 【样例 1 输出】

```
2
```

### 【样例 1 解释】

在第 2 轮攻击后，战舰  $(1, 1, 1)$  总共受到了 2 点伤害，超出其防御力导致爆炸。

**【样例 2】**

见下发样例

**【数据范围】**

对于 10% 的数据,  $B = C = 1$ ;

对于 20% 的数据,  $C = 1$ ;

对于 40% 的数据,  $A \times B \times C \leq 10^4, m \leq 10^4$ ;

对于 70% 的数据,  $A, B, C \leq 200$ ;

对于所有数据,  $A \times B \times C \leq 10^6; m \leq 10^6, 0 \leq d[i][j][k] \leq 10^9; h_t \leq 10^9$   
; 保证所有数据合法。

### 3 填数游戏 game

#### 3.1 题目背景

由于 NOIP2018 出题组的失误, 考场上出现了一道原题. 今年他们由于经费欠缺, 更是发生了这样的闹剧:

当你走进 NOIP2019 考场, 打开题面, 发现竟然只有一句话:

Please contact noi@ccf.org.cn!

机智的你发现了下面还有一行隐藏的字:

哈哈哈哈哈, 笑死你了, 这一笑你就笑醒了过来. 你意识到今天是 NOIP2019, 你坐的车刚开进长沙理工大学. 你把这个梦告诉了周围的同学, 并预计今年又会出原题.

走进考场, 打开真正的题面, 读到第三题填数游戏, 果然你奶中了今年考原题?! 喜欢毒瘤题的你正准备闷声发大财, 又发现这不是去年那道填数游戏了. 于是你认真读起题目来.

#### 3.2 题目描述

在  $n \times m$  的矩阵里填数, 求满足以下要求的方案数

1.  $0 \leq a_{i,j} \leq m$
2. 对于  $1 \leq i \leq n; 1 \leq j < m$ , 满足  $a_{i,j} < a_{i,j+1}$
3. 对于  $1 < i \leq n; 1 \leq j < m$ , 满足  $a_{i,j} < a_{i-1,j+1}$

答案对  $10^9 + 7$  取模.

#### 3.3 输入格式

一行两个整数, 分别为  $n$  和  $m$

#### 3.4 输出格式

一个整数, 表示方案数

### 3.5 样例

#### 3.5.1 game1.in

3 3

#### 3.5.2 game1.out

40

### 3.6 数据范围

对于 20% 的数据,  $n, m \leq 300$

对于 40% 的数据,  $n, m \leq 5000$

对于另外 5% 的数据,  $n = 2$  ; 对于另外 5% 的数据,  $m = 2$

对于 80% 的数据,  $n, m \leq 10^6$

对于 100% 的数据,  $n, m \leq 10^7$