

# 提高组模拟赛

VanishD

August 12, 2019

题目名称	辩论	数独	面积
目录	debate	hexodoku	area
可执行文件名	debate	hexodoku	area
输入文件名	debate.in	hexodoku.in	area.in
输出文件名	debate.out	hexodoku.out	area.out
每个测试点时限	2s	2s	2s
内存限制	512MB	512MB	32MB
试题总分	100	100	100
测试点数目	10	10	10
每个测试点分值	10	10	10
是否有部分分	否	否	否
题目类型	传统型	传统型	传统型

提交的源程序文件名

对于 C++ 语言	debate.cpp	hexodoku.cpp	area.cpp
对于 C 语言	debate.c	hexodoku.c	area.c
对于 Pascal 语言	debate.pas	hexodoku.pas	area.pas

编译开关

对于 C++ 语言	-O2 -lm	-O2 -lm	-O2 -lm
对于 C 语言	-O2 -lm	-O2 -lm	-O2 -lm
对于 Pascal 语言	-O2	-O2	-O2

WARNING: 题目并不难, AK 后请不要大声喧哗。

# 1 辩论

## 1.1 题目描述

一场辩论会开始了，这场辩论会上有两个议题 1,2。

有  $N$  个参加辩论的候选人，每个人对这两个议题都有明确的态度，支持或反对。

作为组织者，小 D 认真研究了每个候选人，并给每个人评估了一个非负的活跃度，他想让活跃度之和尽可能大。

但是，由于一些不可抗力，选出的候选人必须满足以下两个条件：

1. 至少有一半的人支持议题 1（如果是奇数上取整）。
2. 至少有一半的人支持议题 2。

小 D 想知道，在满足以上两个条件的情况下，活跃度之和最大是多少。

## 1.2 输入格式

从文件 `debate.in` 中读取数据。

第一行一个整数  $N$ ，描述辩论会的候选人。

接下来  $N$  行，每行描述一个候选人的态度和活跃度，由一个长度为 2 的 01 串和一个整数  $A_i$  组成。01 串中第  $i$  位为 0 表示反对第  $i$  个议题，为 1 表示支持， $A_i$  描述活跃度。

## 1.3 输出格式

输出到文件 `debate.out` 中。

一行一个整数，表示辩论会最大的活跃度。

## 1.4 样例 1 输入

```
6
11 6
10 4
01 3
00 3
00 7
00 9
```

## 1.5 样例 1 输出

```
22
```

## 1.6 样例解释

选择 1, 2, 3, 6 这四个候选人。

## 1.7 样例 2 输入

```
6
11 19
10 22
00 18
00 29
11 29
10 28
```

## 1.8 样例 2 输出

```
105
```

## 1.9 数据范围和约定

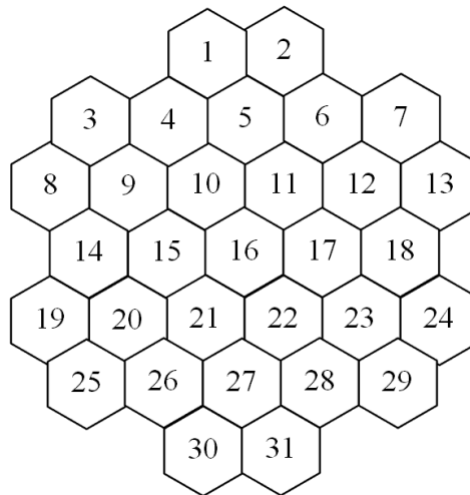
对于 100% 的数据， $N \leq 4 * 10^5$ ， $0 \leq A_i \leq 5 * 10^3$ ，详细数据范围见下表：

测试点编号	$N$	特殊性质
1	$\leq 10$	无
2		
3	$\leq 5000$	
4		
5	$\leq 4 * 10^5$	所有 $A_i$ 相同
6		一个候选人要么同时支持两个议题，要么同时反对
7		
8		
9		无
10		

## 2 数独

### 2.1 题目描述

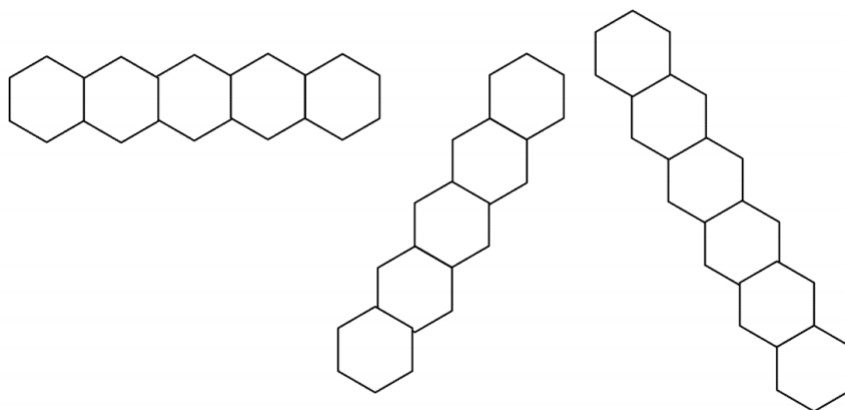
考虑一个不标准的数独模型，它的棋盘如下所示：



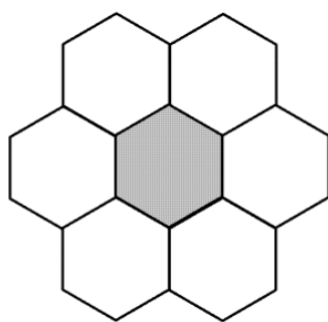
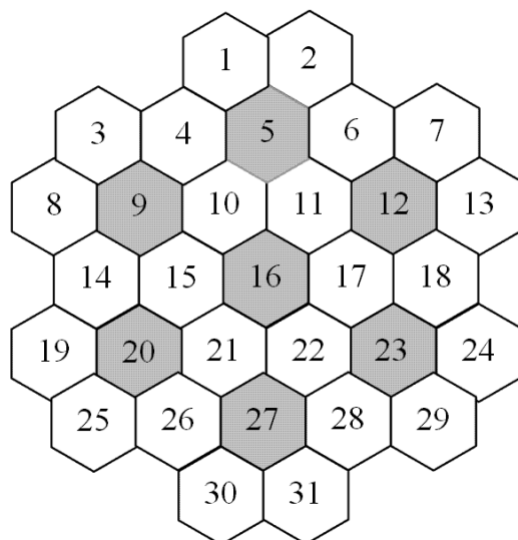
格子从 1 到 31 标号。

根据规则，可以往每个格子中填 1 到  $K$  之间的数，每一行中的数都不能相同。

在这个数独中，以下的情况都视为一行（具体见样例解释 1）：



另外地，对于每个深颜色的格子，它和它相邻的格子中的任意一对格子都不能出现填的数编号相同的情况。



每个格子中的数都必须是不同的。

有些格子中可能已经填好了数，你要找出字典序下第  $N$  小的答案。

保证至少有一组合法解。

## 2.2 输入格式

从文件 `hexodoku.in` 中读取数据。

第一行两个整数  $K, N$ ，意义见题面。

第二行 31 个整数，依次给出数独中每个格子中的数，若是 0 则表示这一格还未确定。

## 2.3 输出格式

输出到文件 `hexodoku.in` 中。

如果存在字典序第  $N$  小的解，输出 "Found"，并且在第二行中按输出数独中的每个数。

如果不存在，输出 "No way"

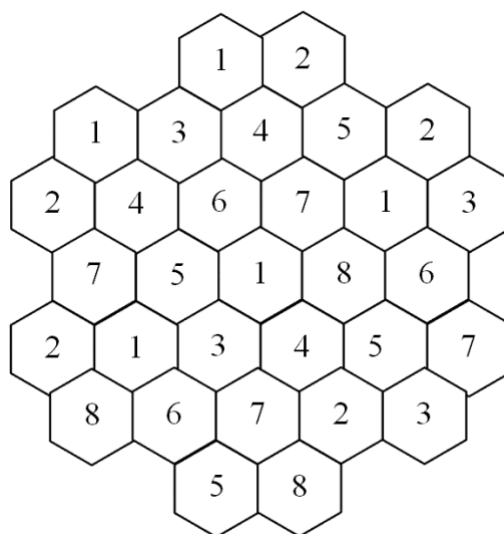
## 2.4 样例 1 输入

```
8 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

## 2.5 样例 1 输出

```
Found
1 2 1 3 4 5 2 2 4 6 7 1 3 7 5 1 8 6 2 1 3 4 5 7 8 6 7 2 3 5 8
```

## 2.6 样例解释



## 2.7 样例 2 输入

```
7 100000
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

## 2.8 样例 2 输出

```
No way
```

## 2.9 样例 3 输入

```
9 10
1 2 0 3 4 5 2 0 4 6 7 1 3 0 5 1 8 6 0 1 3 4 5 0 8 6 7 2 3 5 8
```

## 2.10 样例 3 输出

Found

1 2 1 3 4 5 2 8 4 6 7 1 3 7 5 1 8 6 2 1 3 4 5 9 8 6 7 2 3 5 8

## 2.11 数据范围和约定

对于 20% 的数据,  $K \leq 7$

对于另外 20% 的数据,  $N = 1$

对于另外 20% 的数据,  $K = 31$

对于 100% 的数据,  $k \leq 31, N \leq 100000$

## 2.12 提示

数据在运行时间上有一定梯度。

## 3 面积

### 3.1 题目描述

在数学上，克罗内克积（记作  $\otimes$ ），是一个矩阵间的运算，它的具体定义如下：

定义矩阵  $A, B$ ，大小分别为  $k * l$  和  $n * m$ 。他们的克罗内克积是一个  $kn * lm$  的矩阵  $C$ ：

$$\begin{pmatrix} a_{11}B & a_{12}B & \dots & a_{1l}B \\ a_{21}B & a_{22}B & \dots & a_{2l}B \\ \dots & \dots & \dots & \dots \\ a_{k1}B & a_{k2}B & \dots & a_{kl}B \end{pmatrix}$$

每一个  $a_{ij}B$  代表一个  $m \times n$  的子矩阵：

$$\begin{pmatrix} a_{ij}b_{11} & a_{ij}b_{12} & \dots & a_{ij}b_{1n} \\ a_{ij}b_{21} & a_{ij}b_{22} & \dots & a_{ij}b_{2n} \\ \dots & \dots & \dots & \dots \\ a_{ij}b_{m1} & a_{ij}b_{m2} & \dots & a_{ij}b_{mn} \end{pmatrix}$$

考虑矩阵  $A = \begin{pmatrix} +1 & +1 \\ +1 & -1 \end{pmatrix}$ ，定义  $A^2 = A \otimes A$ ， $A^3 = A^2 \otimes A$ ，...  $A^p = A^{p-1} \otimes A$

最后，我们定义  $A^\infty$  为一个无限的二维数表。可以把这个数表的左上角的数的坐标看作为  $(0,0)$

定义两个数表中的坐标是互相连通的，当且仅当这两个坐标间存在一条路径，使得路径中任意相邻的位置上的数相同且有一条公共边。

小 D 想知道，对于给定的坐标  $(X,Y)$ ，它所在的联通块大小是多少，即有多少个坐标和它连通。

### 3.2 输入格式

从文件 `area.in` 中读取数据。

一行两个整数  $X,Y$ ，表示给定的坐标。

### 3.3 输出格式

输出到文件 `area.out` 中。

输出一行一个整数  $Ans$ ，表示联通快的大小，如果联通块大小是无穷的，输出“infinity”。



### 3.4 样例 1 输入

2 3

### 3.5 样例 1 输出

5

### 3.6 样例解释

矩阵前几行几列为：

```
1  1  1  1  1  1 ...
1 -1  1 -1  1 -1 ...
1  1 -1(-1) 1  1 ...
1 -1 -1  1  1 -1 ...
1  1  1  1  1 -1 ...
.  .  .  .  .  .
.  .  .  .  .  .
.  .  .  .  .  .
```

被 () 标出的即为询问的位置。

它所在的连通块的大小为 5。

### 3.7 样例 2 输入

1 4

### 3.8 样例 2 输出

infinity

### 3.9 数据范围和约定

对于 20% 的数据， $X, Y \leq 100$

对于 50% 的数据， $X, Y \leq 10^4$

对于 100% 的数据， $0 \leq X, Y \leq 10^9$

### 3.10 提示

请注意时间和空间限制。