

# HNOI2019 练习赛

dy0607

March 22, 2019

题目名称	Lemon	Invert	Triplet
每个测试点时限	2s	3s	1s
空间限制	256M	256M	256M
题目类型	传统型	传统型	交互型
编译开关	-lm -O2 -std=c++11		

Note:

1. 评测在Ubuntu 16.04(64bit)上进行, 评测时开启无限栈.
2. 评测机的处理器配置为Intel® Pentium(R) CPU G2030 @ 3.00GHz × 2, 内存4G.
3. 题目难度与顺序无关.

## 1 Lemon

### 1.1 Description

小G在院子里种了一棵柠檬树，这棵树由 $n$ 个节点组成，其中1号节点是根节点，每个节点有一个坚固程度 $d_i$ 。

秋天到了，树上的每个节点都结出了一个柠檬，第 $i$ 个柠檬的重量为 $w_i$ 。然而，如果一个节点的子树中的柠檬的重量的最大值大于该节点的坚固程度，柠檬树就会变得不稳定。

为了使柠檬树变得稳定，小G可以任意交换两个节点上的柠檬（可交换任意次）；除此之外，他还可以对某一个节点进行加固，使这个节点的坚固程度增加 $k$ 。

加固节点是一件费力的工作，小G希望在保证柠檬树稳定的前提下最小化 $k$ 。求出最小的 $k$ ，或判断无解。

### 1.2 Input

从文件`lemon.in`中读入数据。

第一行一个整数 $n$ 。

接下来 $n$ 行，每行三个整数 $d_i, w_i, f_i$ ，其中 $f_i$ 是 $i$ 在柠檬树上的父亲节点。特别地， $f_0 = 0$ 。

### 1.3 Output

输出到文件`lemon.out`中。

如果无解，输出 $-1$ ；如果柠檬树不需要加固就能保持稳定，输出 $0$ ；否则输出最小的 $k$ 。

### 1.4 Sample1

#### 1.4.1 Input

```
6
10 3 0
8 2 1
```

```
3 10 1
7 7 3
7 5 3
2 6 2
```

### 1.4.2 Output

```
3
```

## 1.5 Sample2

### 1.5.1 Input

```
3
100 2 0
1 2 1
1 2 1
```

### 1.5.2 Output

```
-1
```

## 1.6 Sample3-4

见下发文件.

## 1.7 Subtasks

对于所有数据, 有  $1 \leq n \leq 5 \times 10^5, 0 \leq d_i, w_i \leq 10^9. \forall i \in [2, n], 0 < f_i < i.$   
 $f_1 = 0.$

- Subtask1(5%),  $n = 2$ 。
- Subtask2(17%),  $n \leq 7$ 。
- Subtask3(12%),  $f_i = i - 1$ 。

- Subtask4(13%),  $f_i = 1$ 。
- Subtask5(21%),  $n \leq 10^3$ 。
- Subtask6(32%), 没有特殊的限制。

## 2 Invert

### 2.1 Description

Gator是博弈大师。这天Gator觉得很无聊，他找来了他的好朋友Alligator一起研究博弈论。

他们研究的游戏在一个 $n \times n$ 的网格上进行，每个格子要么是黑色，要么是白色。

两人轮流进行操作，Gator先手。每次操作可以选择一个白色格子，将一个以这个格子为右下角且边长不超过 $k$ 的正方形内的所有格子反色。需要保证正方形不超出网格的边界。不能操作者输。

现在给出网格的初始颜色，你需要确定在他们足够聪明的情况下谁能获胜。

### 2.2 Input

从文件`invert.in`中读入数据。

第一行三个整数 $n, m, k$ 。

接下来 $m$ 行，每行四个整数 $a_i, b_i, c_i, d_i$ ，描述一个矩形。记 $(i, j)$ 表示第 $i$ 行第 $j$ 列的格子。那么 $(a_i, b_i)$ 是矩形的左上角， $(c_i, d_i)$ 是矩形的右下角。

如果某个格子被至少一个矩形覆盖，那么这个格子初始为白色，否则为黑色。

### 2.3 Output

输出到文件`invert.out`中。

一行一个字符串，如果先手获胜，输出Gator；否则输出Alligator。

### 2.4 Sample1

#### 2.4.1 Input

```
2 2 1
1 1 2 2
1 2 2 2
```

### 2.4.2 Output

Alligator

## 2.5 Sample2

### 2.5.1 Input

```
2 2 2
1 1 2 2
1 2 2 2
```

### 2.5.2 Output

Gator

## 2.6 Sample3-10

见下发文件.

## 2.7 Subtasks

对于所有数据,  $1 \leq k \leq n \leq 10^9$ ,  $m \leq 5 \times 10^5$ ,  $1 \leq a_i \leq c_i \leq n$ ,  $1 \leq b_i \leq d_i \leq n$ .

- Subtask1(3%),  $m = k = 1$ .
- Subtask2(14%),  $n, m \leq 4$ .
- Subtask3(27%),  $n, m \leq 10^3$ .
- Subtask4(7%),  $m = 1$ .
- Subtask5(19%),  $m \leq 300$ .
- Subtask6(14%),  $m \leq 3 \times 10^4$ .
- Subtask7(16%), 没有特殊的约束.

## 3 Triplet

### 3.1 Description

小G是一名神秘选手，他有一个长为 $n$ 的**元素互不相同**的数组 $A$ (标号为 $1 \rightarrow n$ )，小G的qq号是 $\sum_{i=1}^n A_i \times 233^i \pmod{998244353}$ 。

作为小G的粉丝，你也想知道小G的qq号。但小G并不会直接告诉你，而是让你进行若干次询问，每次询问你可以选择三个**不同**的数 $x, y, z \in [1, n]$ ，小G会回答你 $\max\{A_x, A_y, A_z\} + \min\{A_x, A_y, A_z\}$ 。

由于询问过多次后小G就会变得不耐烦，你需要在尽量少的询问次数内猜出小G的qq号。

### 3.2 Interaction

你需要在`triplet.cpp`中包含`triplet.h`，并实现函数：

```
int guess(int n, int Taskid);
```

其中 $n$ 为数组长度， $Taskid$ 是子任务编号。函数的返回值应当为小G的qq号。注意本题有多组数据(数据组数 $T \leq 10$ )，因此交互库会对这个函数进行多次调用。

你可以调用接口：

```
int query(int x, int y, int z);
```

来对小G进行询问，其意义见题目描述。注意如果你给出的参数不合法，将出现Runtime Error.

### 3.3 Notes

你可以使用

```
g++ grader.cpp triplet.cpp -o grader -O2 -std=c++11
```

来对程序进行编译。编译得到的可执行文件的输入输出格式如下：

### 3.3.1 Input

从标准输入读入数据。

第一行三个整数 $T, Taskid, Maxscore$ ，表示数据组数，子任务编号，和测试点分值。

接下来有 $T$ 组数据，每组数据第一行一个整数 $n$ ，第二行 $n$ 个整数表示 $A$ 数组。

### 3.3.2 Output

输出到标准输出中。

若选手猜错，将输出`wa x`，其中 $x$ 为猜错的数据编号。

否则将输出`ok S`，其中 $S$ 表示选手在该测试点获得得分。

### 3.4 Grading

对于某组数据，设 $P = \min\{n + 35, 2 \times n\}$ ，而你使用了 $x$ 次询问猜出序列，那么你的得分 $S$ 为：

$$S = Maxscore \times \begin{cases} 1 & x \leq P \\ 0.9 - 0.6 \times \frac{x-P}{P} & P < x \leq 1.5 \times P \\ 0.7 - 0.2 \times \frac{x-P}{P} & 1.5 \times P < x \leq 3 \times P \\ \max\{0.1, 0.34 - 0.02 \times \frac{x-P}{P}\} & x > 3 \times P \end{cases}$$

一个测试点的得分为各组数据得分的最小值。任意一次猜错将导致测试点得0分。

### 3.5 Hints

- 请不要在你实现的函数中进行输入输出，否则会导致得0分。
- 请不要使用除了`query()`外的其他交互库中的变量或函数，最终评测时使用的`grader.cpp`将有所不同。



- 保证交互库占用的空间不超过10M，除询问以外占用的时间不超过0.1s，而询问占用的时间由你的询问次数而定（评测机上 $5 \times 10^7$ 次询问占用时间不超过1s）。
- 交互库不是adaptive的，即A序列是初始确定的，不会因为你的询问而发生改变。

### 3.6 Subtasks

对于所有数据，有 $1 \leq T \leq 10, 1 \leq Taskid \leq 5, 5 \leq n \leq 10^4, 1 \leq A_i \leq 10^9$ ， $A_i$ 两两不同。数据保证你一定能猜出小G的qq号。

小G可能有很多小号，所以每次的答案不一定相同。

Subtask0(0%), 答案是小G的qq号。

- Subtask1(14%),  $A_1 = 1, A_2 = 2$ .
- Subtask2(24%),  $A_1 = 1, A_2 = 10^9$ .
- Subtask3(20%),  $A_1 = 2 \times 10^8, A_2 = 5 \times 10^8, A_3 = 8 \times 10^8$ .
- Subtask4(27%),  $n = 5$ .
- Subtask5(15%), 没有特殊的约束。