

# 全国青少年信息学奥林匹克联赛

## 提高组（复赛）

### 模拟赛

2019 年 9 月 2 日

8:30 - 12:00

#### 一、题目概况

中文题目名称	Dove 爱旅游	Cicada 爱烧烤	Dove 的博弈
英文题目与子目录名	trip	string	game
可执行文件名	trip	string	game
输入文件名	trip.in	string.in	game.in
输出文件名	trip.out	string.out	game.out
每个测试点时限	1s	1s	3s
内存上限	128MB	128MB	256MB
测试点数目	20	25	20
每个测试点分值	5	4	5
附加样例文件	有	有	有
结果比较方式	全文比较（过滤行末空格及文末回车）		
题目类型	传统	传统	传统

#### 二、提交源程序程序名

对于 C++ 语言	.cpp	.cpp	.cpp
对于 C 语言	.c	.c	.c

### 三、优化开关

对于 C++ 语言	-O2 -std=c++11	-O2 -std=c++11	-O2 -std=c++11
对于 C 语言	-O2 -std=c11	-O2 -std=c11	-O2 -std=c11

### 四、评测说明

1. 所有题目的时限是在CPU: Intel i5-7360U (4) @ 2.30GHz的运行条件下给出, 数据制作以及标算的编写是在OS: macOS Mojave 10.14.5 18F132 x86\_64中进行, 所有程序均采用GCC@9.1.0进行编译。
2. 保证每道题目标算的运行时间不超过给定时限的 60%。
3. 本场比赛的所有题目均支持 C++11, 并且开启 O2 优化。

# 1 Dove 爱旅游

(trip.cpp/c)

## 1.1 问题描述

住在 C 国的 Dove 喜爱游览祖国的大好河山，他经常会为自己规划旅游路线。

C 国有  $n$  个城市，有  $n - 1$  条道路连接着这些城市，对于每一条道路，保证其连接的是两个不同的城市，同时保证任意两条道路的两端不会同时连接相同的城市。每个城市都会有一个著名景点，可以是湖泊或者山川。对于第  $i$  个城市，如果  $a_i = 0$ ，则表示这个城市的著名景点是湖泊，如果  $a_i = 1$ ，则表示这个城市的著名景点是山川。

Dove 每次旅行前都会规划一个旅行路线，一个合法的旅行路线表示为这  $n$  个城市组成的图的一个子联通块。Dove 讨厌浏览相同种类的景点，所以对于一个旅游路线来说，Dove 定义其无聊度为旅行路线中湖泊与山川数量差值的绝对值。

Dove 想知道在所有可能的旅游路线中，无聊度最大的旅行路线的无聊度是多少。

## 1.2 输入

第一行输入一个整数  $n$ ，表示 C 国的城市数。

接下来一行  $n$  个整数，第  $i$  个数为  $a_i$ 。

最后  $n - 1$  行，每行 2 个整数，表示 C 国中的道路。

## 1.3 输出

输出一行一个整数，表示无聊度最大的旅行路线的无聊度。

## 1.4 样例

sample/trip\*.in  
sample/trip\*.out

## 1.5 约定和数据范围

对于全部测试点，保证  $n \leq 10^6, a_i \in \{0, 1\}$ 。

测试点	$n$	特殊性质
1, 2, 3, 4, 5	$\leq 20$	
6, 7, 8	$\leq 500$	
9, 10	$\leq 3000$	保证 $a_i$ 相同
11, 12, 13	$\leq 3000$	
14, 15, 16, 17, 18, 19, 20		

## 2 Cicada 爱烧烤

(string.cpp/c)

### 2.1 问题描述

Cicada 喜爱烧烤,有很多食材都可以用来烧烤食用,常见的有羊肉串、牛肉串。不过 Cicada 最喜欢吃的还是字符串中的回文串!

对于一个字符串  $s$  来说,我们定义其反串  $s^t$  为  $s$  前后倒置构成的新字符串。例如  $s = "abcc"$ , 那么  $s^t = "ccba"$ 。对于一个字符串来说,如果满足  $s = s^t$ , 那么我们称  $s$  是一个「回文串」。

同时, 对于一个字符串  $s$  来说, 我们定义其第  $i$  个前缀为  $s$  的前  $i$  个字符构成的字符串。Cicada 喜爱回文串, 但是他讨厌一个回文串中还有前缀是回文串。Cicada 认为一个长度为  $n$  的字符串是可爱的, 当且仅当这个字符串是一个回文串, 并且其任意的第  $i$ , ( $2 \leq i \leq n-1$ ) 个前缀不是回文串。

现在 Cicada 想知道, 对于所有可能存在的长度为  $n$ , 字符集大小为  $m$  的字符串来说, 有多少个字符串是可爱的。你需要计算答案并输出对  $10^9 + 7$  取模的结果。

### 2.2 输入

一行两个整数  $n, m$ , 为字符串长度和字符集的大小。

### 2.3 输出

一行一个整数, 表示答案对  $10^9 + 7$  取模的结果。

### 2.4 样例

```
sample/string*.in
sample/string*.out
```

## 2.5 约定和数据范围

对于全部测试点，保证  $n \leq 10^6, m \leq 10^9$ 。

测试点	$n$	$m$
1, 2, 3, 4	$\leq 5$	$\leq 5$
5, 6, 7, 8	$\leq 20$	$\leq 2$
9, 10, 11, 12	$\leq 3000$	$\leq 3000$
13, 14, 15, 16	$\leq 3000$	$\leq 10^9$
17, 18, 19, 20	$\leq 10^5$	$\leq 10^5$
21, 22, 23	$\leq 10^6$	$\leq 10^6$
24, 25		

## 3 Dove 的博弈

(game.cpp/c)

### 3.1 问题描述

Dove 喜爱博弈问题，经常和 Cicada 一起玩游戏。

一天 Dove 和 Cicada 在玩这样的游戏。有  $n$  个棋子排成一排，每个棋子都有一个权值，其中从左向右第  $i$  个棋子的权值为  $a_i$ 。游戏是这样进行的，我们提前给定一个常数  $m$ ，Dove 和 Cicada 交替操作棋子。每次操作者可以任意选择一个位置上的棋子，假设其选择的是第  $i$  个棋子，权值为  $a_i$ 。那么他可以选择吃掉一个相邻的满足权值  $a_j$  不超过  $a_i + m$  的棋子  $j$ ，使得棋子  $i$  的权值增加  $a_j$ ，同时删去棋子  $j$ ，不能操作者败。

形式化的，对于任意一次操作来说，如果选择了  $i$ ，那么必须选择一个  $j \in \{i-1, i+1\}, a_j \leq a_i + m$ ，使得  $a_i \leftarrow a_i + a_j$  并且消除  $a_j$ 。

Dove 和 Cicada 并不足够聪明，所以他们并不关心是否存在必胜策略，他们只关心在随机执行的前提下，哪些棋子能够留在最后。

### 3.2 输入

第一行两个整数  $n, m$ ，表示棋子数量和给定的常数。

接下来一行  $n$  个整数，第  $i$  个数表示最初第  $i$  个棋子的权值。

### 3.3 输出

输出一行若干个递增的整数，表示能够留下的棋子的标号。

### 3.4 样例

```
sample/game*.in  
sample/game*.out
```

### 3.5 约定和数据范围

对于全部测试点，保证  $n \leq 8 \times 10^6, a_i, m \leq 10^9$ 。

测试点	$n$
1, 2, 3	$\leq 20$
4, 5, 6, 7	$\leq 3000$
8, 9, 10	$\leq 5 \times 10^4$
11, 12, 13	$\leq 10^5$
14	$\leq 5 \times 10^5$
15	$\leq 10^6$
16, 17, 18, 19, 20	$\leq 8 \times 10^6$