

# 全国青少年信息学奥林匹克竞赛

## NOIP2023模拟

时间：8:00-12:20

题目名称	牧场施肥	堆叠纸张	饥饿奶牛	观看牛飞
题目类型	传统型	传统型	传统型	传统型
目录	fertilizing	papers	cow	cowflix
可执行文件名	fertilizing	papers	cow	cowflix
输入文件名	fertilizing.in	papers.in	cow.in	cowflix.in
输出文件名	fertilizing.out	papers.out	cow.out	cowflix.out
每个测试点时限	1.0秒	1.0秒	2.0秒	2.0秒
内存限制	256 MB	256 MB	512MB	256MB
子任务数目	20	10	20	20
测试点是否等分	是	是	是	是

### 提交源程序文件名

对于C++语言	fertilizing.cpp	papers.cpp	cow.cpp	cowflix.cpp
---------	-----------------	------------	---------	-------------

### 编译选项

对于C++语言	-lm -std=c++14 -O2
---------	--------------------

### 注意事项与提醒（请选手务必仔细阅读）

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C++ 中主函数的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
3. 提交的程序代码文件的放置位置请参照各省的具体要求。
4. 因违反以上三点而出现的错误或问题，申诉时一律不予受理。
5. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
6. 程序可使用的栈内存空间限制与题目的内存限制一致。
7. 全国统一评测时采用的机器配置为：Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz，内存 32GB。上述时限以此配置为准。
8. 评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以其为准。
9. 终评测时所用的编译命令中不含编译选项之外的任何优化开关。

# 牧场施肥 (fertilizing)

## 【问题描述】

有  $N$  个牧场 ( $2 \leq N \leq 2 \cdot 10^5$ )，由  $N - 1$  条道路连接，形成一棵树。每一条路需要花费 1 秒的时间通过。每个牧场从 0 棵草开始，第  $i$  个牧场的草以每秒  $a_i$  ( $1 \leq a_i \leq 10^8$ ) 个单位的速度生长。农夫约翰一开始就在 1 号牧场，他需要开车四处走动，给每个牧场的草施肥。如果他去一个有  $x$  单位草的牧场，它将需要  $x$  个肥料。牧场只需要在第一次到访时施肥，而给牧场施肥需要 0 分钟。

输入包含一个额外的参数  $T \in \{0, 1\}$ 。

- 如果  $T = 0$ ，农夫约翰必须在牧场 1 结束。
- 如果  $T = 1$ ，农夫约翰可以在任何牧场结束。

计算给每个牧场施肥所需的最短时间，以及在这段时间内完成施肥所需要的最短肥料量。

## 【输入格式】

第一行包含  $N$  和  $T$ 。

第 2 行到第  $N$  行，包含两个整数  $p[i]$  和  $a[i]$ ， $a[i]$  的含义见上文。 $p[i]$  则表示节点  $i$  和  $p[i]$  之间有一条边相连。

## 【输出格式】

输出包含两个整数：遍历所有节点的最小时间和此时需要付出的费用。

## 【样例输入1】

```
5 0
1 1
1 2
3 1
3 4
```

## 【样例输出1】

```
8 21
```

## 【样例1解释】

农夫约翰最佳路线如下：

在时间 1，移动到节点 3，节点 3 现在有  $1 * 2 = 2$  棵草，因此需要 2 个肥料。

在时间 2，移动到节点 5，该节点现在有  $2 * 4 = 8$  棵草，因此需要 8 个肥料。

在时间 3，移回节点 3，我们已经施肥了，所以不需要再次施肥。

在时间 4，移动到节点 4，节点 4 现在有  $4 * 1 = 4$  棵草，因此需要 4 个肥料。

在时间 5，移回节点 3，我们已经施肥了。

在时间 6，移回节点 1。

在时间 7，移动到节点 2，节点 2 现在有  $7 * 1 = 7$  棵草，因此需要 7 个肥料。

在时间 8，返回到节点 1。

该路线耗时 8 次，使用  $2 + 8 + 4 + 7 = 21$  个肥料。可以看出，8 是在末端返回到节点 1 的任何路由的最小可能的时间量，而 21 是返回节点 1 并花费 8 个时间的任何路由使用的最小可能肥料。

### 【样例输入2】

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

### 【样例输出2】

```
6 29
```

### 【样例2解释】

农夫约翰的最佳路线如下：

在时间 1，移动到节点 2，该节点现在有  $1 * 1 = 1$  棵草，因此需要 1 个肥料。

在时间 2，移回节点 1。

在时间 3，移动到节点 3，节点 3 现在有  $3 * 2 = 6$  棵草，因此需要 6 个肥料。

在时间 4，移动到节点 5，该节点现在有  $4 * 4 = 16$  棵草，因此需要 16 个肥料。

在时间 5，移回节点 3，我们已经施肥了，所以不需要再次施肥。

在时间 6，移动到节点 4，节点 4 现在有  $6 * 1 = 6$  棵草，因此需要 6 个肥料。

这条路线需要 6 个时间，使用  $1 + 6 + 16 + 6 = 29$  肥料。可以看出，对于任何路线而言，6 是尽可能少的时间量，而对于任何花费 6 个时间的路线而言，29 是尽可能最少的肥料。

### 【数据范围及约定】

测试点 1 ~ 8，满足  $T = 0$

测试点 9 ~ 20，满足  $T = 1$

另外测试点 1 ~ 4，9 ~ 12，是一条链。

# 堆叠纸张 (papers)

## 【问题描述】

农夫约翰在一张纸上写下  $N$  ( $1 \leq N \leq 300$ ) 个数字。对于  $[1, N]$  中的每个  $i$ , 第  $i$  张纸包含数字  $a_i$  ( $1 \leq a_i \leq 9$ )。

奶牛有两个最喜欢的整数  $A$  和  $B$  ( $1 \leq A \leq B < 10^{18}$ )，希望您回答  $Q$  ( $1 \leq Q \leq 5 \times 10^4$ ) 查询。对于第  $i$  个查询，奶牛将在纸张  $l_i \cdots r_i$  ( $1 \leq l_i \leq r_i \leq N$ ) 上从左到右移动，保持最初的空纸张堆。对于每一张纸，他们要么把它加到纸堆的顶部，要么加到纸堆底部，要么两者都不加。最后，他们将从上到下阅读这堆论文，形成一个整数。在奶牛在此过程中做出选择的所有  $3^{r_i - l_i + 1}$  种方式中，计算最好奶牛读取数值在  $[A, B]$  (包括  $[A, B]$ ) 不同的操作方案数，对  $10^9 + 7$  取模。

## 【输入格式】

第一行包含三个空格分隔的整数  $N, A$  和  $B$ 。

第二行包含  $N$  个空格分隔的数字  $a_1, a_2, \dots, a_N$ 。

第三行包含一个整数  $Q$ ，即查询数。

接下来的  $Q$  行分别包含两个空间分隔的整数  $l_i$  和  $r_i$ 。

## 【输出格式】

对于每个查询，输出一行表示答案。

## 【样例输入1】

```
5 13 327
1 2 3 4 5
3
1 2
1 3
2 5
```

## 【样例输出1】

```
2
18
34
```

## 【样例1解释】

对于第一个查询，贝茜在阅读区间  $[1, 2]$  时可以通过九种方式堆叠纸张：

- 贝茜可以忽略 1，然后忽略 2，得到 0。
- 贝茜可以忽略 1，然后在堆栈顶部添加 2，得到 2。
- 贝茜可以忽略 1，然后在堆栈底部添加 2，得到 2。
- 贝茜可以在堆栈顶部加 1，然后忽略 2，得到 1。
- 贝茜可以在堆栈顶部加 1，然后在堆栈顶部再加 2，得到 21。
- 贝茜可以在堆栈顶部加 1，然后在堆栈底部加 2，得到 12。
- 贝茜可以在堆栈底部加 1，然后忽略 2，得到 1。
- 贝茜可以在堆栈底部加 1，然后在堆栈顶部加 2，得到 21。

- 贝茜可以在堆栈底部加 1，然后在堆栈底部再加 2，得到 12。

只有给出 21 的两种方式才能得到 13 到 327 之间的数字，所以答案是 2。

### 【样例2】

见下发样例文件中。

### 【数据范围及约定】

对于测试点 1 ~ 2，满足  $B < 100$ ；

对于测试点 3 ~ 4，满足  $A = B$ ；

对于测试点 5 ~ 10，无额外限制条件。

# 饥饿奶牛 (cow)

## 【问题描述】

Bessie 是一头很饿的奶牛。每天晚饭如果谷仓里有干草的话，她会吃一捆干草。FJ 不想让 Bessie 饿着，所以在一些天他会送一些干草，这些干草会早上送到（在晚饭之前）。具体来说，在第  $d_i$  ( $1 \leq d_i \leq 10^{14}$ ) 天，FJ 会送  $b_i$  ( $0 \leq b_i \leq 10^9$ ) 捆干草。

处理  $U$  ( $1 \leq U \leq 10^5$ ) 次如下的更新操作：给定一个数对  $(d, b)$ ，将第  $d$  天送到的干草捆数更新为  $b$ 。在每次更新后，输出 Bessie 所有能吃到草的日期之和对  $10^9 + 7$  取模后的值。

## 【输入格式】

第一行一个整数  $U$ 。

接下来  $U$  行，每行一次更新操作。

## 【输出格式】

输出每次更新操作之后的日期和对  $10^9 + 7$  取模后的结果。

## 【样例输入1】

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

## 【样例输出1】

```
15
36
18
```

## 【样例1解释】

每次更新后的答案是：

1.  $4 + 5 + 6 = 15$

2.  $1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 = 36$

3.  $1 + 2 + 4 + 5 + 6 = 18$

## 【样例输入2】

```
9
1 89
30 7
101 26
1 24
5 1
60 4
5 10
101 0
1 200
```

## 【样例输出2】

```
4005
4656
7607
3482
3507
3753
4058
1107
24531
```

## 【数据范围及约定】

测试点 1, 满足  $U \leq 5\,000$

测试点 2 ~ 8, 满足更新只会增加第  $d$  天送到的干草

测试点 9 ~ 20, 无附加限制

# 观看牛飞 (cowflix)

## 【问题描述】

Bessie 喜欢看牛飞上的视频，并且她会在一些不同的地方看视频。FJ 的农场可以用一棵有  $N$  ( $2 \leq N \leq 2 \cdot 10^5$ ) 个节点的树来表示，并且对于每个节点，要么 Bessie 在这里看牛飞，要么不看。保证 Bessie 至少在一个节点处看牛飞。

不幸的是，牛飞推出了一个新的订阅模式以打击密码共享。在这个新模式中，你可以选择农场中一个大小为  $d$  的连通分量，支付  $d + k$  块钱后，这个账号才能在这个连通分量中使用。形式化地说，你需要选择一组互不相交的连通分量  $c_1, c_2, \dots, c_C$  使得每个 Bessie 会在其上看牛飞的节点包含在某个  $c_i$  中。这组连通分量的花费为

$\sum_{i=1}^C (|c_i| + k)$ ，其中  $|c_i|$  指连通分量  $c_i$  中的节点个数。Bessie 不会在其上看牛飞的节点不需要包含在任何  $c_i$  中。

Bessie 担心新的订阅模式对于她会到的地方来说太贵了，所以她在想要不要改用哔哩哔哩。为了帮助她进行决策，请计算如果保持她的浏览习惯的话，使用牛飞最少要支付多少钱。因为牛飞并未公布  $k$  的值，请计算对于  $k$  取从 1 到  $N$  的所有整数时的答案。

## 【输入格式】

第一行一个整数  $N$ 。

第二行一个二进制串  $s_1 s_2 s_3 \dots s_N$ ，其中如果 Bessie 会在节点  $i$  处看牛飞的话， $s_i = 1$ 。

接下来  $N - 1$  行，每行两个整数  $a, b$  ( $1 \leq a, b \leq N$ )，表示树上一条  $a$  和  $b$  之间的边。

## 【输出格式】

输出  $N$  行，第  $i$  行表示  $k = i$  时的答案。

## 【样例输入1】

```
5
10001
1 2
2 3
3 4
4 5
```

## 【样例输出1】

```
4
6
8
9
10
```

## 【样例1解释】

对于  $k \leq 3$ ，最优的情况是有两个账号： $c_1 = \{1\}, c_2 = \{5\}$ 。对于  $k \geq 3$ ，最优的情况是有一个账号： $c_1 = \{1, 2, 3, 4, 5\}$ 。



### 【样例输入2】

```
7
0001010
7 4
5 6
7 2
5 1
6 3
2 5
```

### 【样例输出2】

```
4
6
8
9
10
11
12
```

### 【数据范围及约定】

测试点 1 ~ 3, 满足  $N \leq 5\,000$ ;

测试点 4 ~ 6, 满足对于所有  $i \in [1, N)$ ,  $i$  与  $i + 1$  有边相连

测试点 7 ~ 15, 满足  $N \leq 10^5$ ;

测试点 16 ~ 20, 无附加限制