

# NOIP 2018

## Day 2

dy0607

August 1, 2018

题目名称	Lighthouse	Miner	Revive
源文件名	lighthouse	miner	revive
输入文件名	lighthouse.in	miner.in	revive.in
输出文件名	lighthouse.out	miner.out	revive.out
题目类型	传统型	传统型	传统型
每个测试点时限	1.0s	0.5s	1.0s
空间限制	512MB	512MB	512MB
编译命令	-lm -O2 -std=c++11		

Notes:

1. 评测在Ubuntu16.04(64bit)上进行，评测时开启无限栈；
2. 评测机配置为Intel® Pentium(R) CPU G2030 @ 3.00GHz × 2，内存4GB；
3. 遇到原题请不要大喊“这不是xx上的xx题吗”，可以AK后提前离场；
4. 题目难度可能与顺序无关。

# 1 Lighthouse

Billions of lighthouses...stuck at the far end of the sky.

## 1.1 Description

平面有 $n$ 个灯塔，初始时两两之间可以相互交流；但由于地形原因，有 $m$ 对灯塔之间无法进行直接的交流。也就是一张完全图缺少了 $m$ 条边。

River想把这 $n$ 个灯塔连成一个环，使得 $n$ 个灯塔都在环上，并且环上相邻的两个灯塔能进行直接交流。River想知道这样做的方案数是多少，两种方案被认为是不同的，当且仅当有两个灯塔 $u, v$ ，它们在一种方案中在环上相邻，而在另一种方案中相反。

答案可能很大，你只需要输出对 $10^9 + 7$ 取模的结果。

## 1.2 Input

从文件 *lighthouse.in* 中读入数据。

第一行两个整数 $n, m$ 。

接下来 $m$ 行，每行描述一条缺少的边。

## 1.3 Output

输出到文件 *lighthouse.out* 中。

一行一个整数表示答案。

## 1.4 Sample1

### 1.4.1 Input

4 1

1 2

### 1.4.2 Output

1

### 1.4.3 Explanation

唯一的方案是(1, 3, 2, 4)依次连成环。

## 1.5 Sample2

### 1.5.1 Input

```
10 3
1 9
3 8
2 7
```

### 1.5.2 Output

```
87840
```

## 1.6 Sample3

### 1.6.1 Input

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

### 1.6.2 Output

```
905583295
```

## 1.7 Subtasks

对于所有数据，有  $3 \leq n \leq 10^7$ ,  $0 \leq m \leq \min(20, \frac{n(n-1)}{2})$ . 输入的边中没有重边和自环

子任务编号	$n$	$m$	分值
1	$\leq 8$	$\leq 20$	17
2	$\leq 18$		14
3	$\leq 50$	$\leq 8$	26
4	$\leq 10^7$	$= 0$	5
5		$= 1$	7
6		$= 2$	9
7		$\leq 20$	22

## 2 Miner

And the universe said you are the daylight  
And the universe said you are the night  
And the universe said the darkness you fight is within you  
And the universe said the light you seek is within you

### 2.1 Description

在Minecraft中，小C最喜欢做的工作就是挖矿，但他从不自己挖矿洞，因为这样每挖不久就会遇到铁矿挡在面前，而小C是不屑于打铁的；于是他总是利用天然矿洞，但这样很容易迷路或者走过重复的路径，降低了他收割Au的效率，于是他希望对路径进行规划。

在小C眼里，矿洞是一个 $n$ 个点的无向图，其中有 $m$ 条无向通道连接。小C可以从任选一个起点开始，进行以下操作中的一个：

- 0  $v$ , 选择一条相邻的还未经过的通道走到 $v$
- 1  $v$ , 利用修改器，直接传送到任意一个点 $v$

小C希望在经过所有通道恰好一次的前提下，最小化传送的次数。输出这个最小次数，并给出任意一个最优方案。

### 2.2 Input

从文件`miner.in`中读入数据。

第一行两个整数 $n, m$ ，接下来 $m$ 行，每行两个整数描述一条无向边。

### 2.3 Output

输出到文件`miner.out`中。

首先输出一个整数表示答案 $ans$ 。

接下来一行，输出你选择的起点 $S$ 。

接下来 $ans + m$ 行，每行两个整数描述一个操作，格式见问题描述。

### 2.4 Sample1

#### 2.4.1 Input

```
5 4
1 2
```

1 2  
3 4  
5 5

### 2.4.2 Output

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

### 2.4.3 Explanation

请注意方案不唯一。

## 2.5 Sample2

见选手目录下的 *miner/miner2.in* 与 *miner/miner2.ans*。

## 2.6 Sample3

见选手目录下的 *miner/miner3.in* 与 *miner/miner3.ans*。

## 2.7 Scoring

如果你答案的第一行正确，你能拿到该测试点40%的分数（向下取整）。一个 *Subtask* 的得分是所有测试点得分的最小值。

## 2.8 Subtasks

对于所有数据，有  $1 \leq n, m \leq 10^5$ 。图中可能出现重边和自环，图也可能不连通。

子任务编号	$n$	$m$	特殊性质	分值
1	$\leq 2$	$\leq 2$	无	7
2	$\leq 6$	$\leq 6$		15
3	$\leq 15$	$\leq 15$		12
4	$\leq 100$	$\leq 10^3$		23
5	$\leq 10^3$	$\leq 10^4$	答案为0	17
6	$\leq 10^5$	$\leq 10^5$	无	26

### 3 Revive

Sparkling ashes drift along your flames  
And softly merge into the sky

#### 3.1 Description

一棵 $n$ 个点的树，每条边都有一个生机值 $w_i$ 。定义 $liveliness(u, v)$ 为 $u$ 到 $v$ 的路径上边的生机值的和。那么整棵树的生机值为：

$$\sum_{u=1}^n \sum_{v=u+1}^n liveliness^2(u, v)$$

你会进行 $q$ 次操作，每次将一条边的生机值增加某个值；在所有操作前以及每次操作后，输出整棵树的生机值。

由于答案可能很大，输出它对 $10^9 + 7$ 取模的结果。

#### 3.2 Input

从文件`revive.in`中读入数据。

第一行一个整数 $id$ 表示子任务编号。

接下来一行两个整数 $n, q$ 。

接下来 $n - 1$ 行，第 $i$ 行两个整数 $f_{i+1}, w_{i+1}$ ，分别表示树上 $i + 1$ 的父亲，以及到父亲的边的生机值。

接下来 $q$ 行，每行两个整数 $u, addw$ ，表示把 $u$ 到 $u$ 的父亲的边的生机值加上 $addw$ 。

#### 3.3 Output

输出到文件`revive.out`中。

第一行一个整数，表示所有操作前这棵树的生机值。

接下来 $q$ 行，第 $i$ 行一个整数表示第 $i$ 个操作后整棵树的生机值。

#### 3.4 Sample1

##### 3.4.1 Input

```
3
5 3
1 3
```

1 4  
2 2  
2 1  
2 7  
3 3  
4 2

### 3.4.2 Output

274  
1072  
1402  
1562

### 3.5 Sample2

见选手目录下的 *revive/revive2.in* 与 *revive/revive2.ans*.

### 3.6 Sample3

见选手目录下的 *revive/revive3.in* 与 *revive/revive3.ans*.

### 3.7 Subtasks

对于所有数据，有  $1 \leq n \leq 10^5, 0 \leq q \leq 5 \times 10^5, 0 \leq w_i, addw \leq 10^9, 1 \leq f_i < i$ ,

子任务编号	$n$	$q$	特殊性质	分值
1	$\leq 100$	$= 0$	无	13
2	$\leq 4000$			5
3				$\leq 4000$
4	$\leq 10^5$	$= 0$	$f_i = i - 1$	8
5		$\leq 10^5$		18
6		$= 0$	无	21
7		$\leq 5 \times 10^5$		13