

# NOIP2024模拟赛

day1

时间：2024 年 8 月 13 日 08:00~12:00

|         |                    |          |          |        |
|---------|--------------------|----------|----------|--------|
| 题目名称    | our                | mask     | tale     | box    |
| 源程序     | our                | mask     | tale     | box    |
| 输入文件名   | our.in             | mask.in  | tale.in  | box.in |
| 输出文件名   | our.out            | mask.out | tale.out | box.in |
| 每个测试点时限 | 3.0s 秒             | 2.0s 秒   | 4.0s 秒   | 1.0 秒  |
| 内存限制    | 512MB              | 512MB    | 512MB    | 512MB  |
| 试题总分    | 100                | 100      | 100      | 100    |
| 子任务数目   | 10                 | 10       | 20       | 20     |
| 测试点是否等分 | 是                  | 是        | 是        | 是      |
| 是否有部分分  | 有                  | 有        | 有        | 有      |
| 题目类型    | 传统型                | 传统型      | 传统型      | 传统型    |
| 比较方式    | 全文比较               | 全文比较     | 全文比较     | 全文比较   |
| 编译开关    | -lm -O2 -std=c++11 |          |          |        |

提交源程序文件名

|           |         |          |          |         |
|-----------|---------|----------|----------|---------|
| 对于 C++ 程序 | our.cpp | mask.cpp | tale.cpp | box.cpp |
|-----------|---------|----------|----------|---------|

## 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/C++ 中函数 main() 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. 只提供 Linux 格式附加样例文件。
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。

# our

## 【题目描述】

*Sea,*

*You,*

*And me.*

*Our lasting memory.*

小  $y$  和他的女朋友小  $x$  在某个城市旅游，这个城市是由  $n$  个餐馆和  $m$  个景区总共  $n + m$  个地点和若干条有向道路连接组成，其中道路只会连接餐馆和景区，餐馆之间和景区之间都没有道路。也就是说，这个城市是一个有向二分图。并且保证其中每个地点都至少有一条向外的道路连到另外一个地点。

小  $y$  和小  $x$  希望在这个城市一直游玩。每到一个地点，如果这是一个餐馆，那么小  $y$  会选择一条从这个餐馆向外的道路走到一个景区；相反如果当前是景区，那么小  $x$  会选择一条从这个景区向外的道路走到一个餐馆。他俩会轮流做选择，一直游玩不停（因为每个点都有出边，所以永远不会在某个点停下，当然这可能会重复经过某些地点，但这对于情侣来说并不重要）。

在开始游玩之前，小  $y$  预先对所有地点都做了调查，知道哪些地点会有好的体验，而剩下的全部都是差的体验，他会尽量避免之后的游玩去到差的地点，但小  $x$  并不在意，所以小  $x$  每次选择会随意选择下一个地点。

现在小  $y$  想知道从哪些地点开始旅游，他通过他的选择（无论小  $x$  如何选择）能够使得他们一直访问到的所有地点（包括起点）都会有好的体验。

## 【输入格式】

从文件 *our.in* 中读取数据。

第一行四个正整数  $n, m, k_0, k_1$  ( $1 \leq n + m \leq 10^6, 1 \leq k_0 + k_1 \leq 10^6$ )，分别表示餐馆的个数，景区的个数，从餐馆走向景区的边数，从景区走到餐馆的边数。

接下来的  $k_0$  行，每行包含两个整数  $x_i, y_i$  ( $0 \leq x_i < n, 0 \leq y_i < m$ )，表示从餐馆  $x_i$  到景区  $y_i$  的有向边。同样的有向边可能出现多次。

接下来的  $k_1$  行，每行包含两个整数  $x_i, y_i$  ( $0 \leq x_i < m, 0 \leq y_i < n$ )，表示从景区  $x_i$  到餐馆  $y_i$  的有向边。同样的有向边可能出现多次。

接下来一行包含两个整数  $n_s, m_s$  ( $0 \leq n_s \leq n, 0 \leq m_s \leq m$ )，表示好的体验的餐馆的数量和好的体验的景区的数量。

接下来一行包含  $n_s$  个两两不同的整数  $ns_i$  ( $0 \leq ns_i < n$ )，表示好的体验的餐馆的编号。

接下来一行包含  $m_s$  个两两不同的整数  $ms_i$  ( $0 \leq ms_i < m$ )，表示好的体验景区的编号。

**【输出格式】**

输出到文件 *our.out* 中。

输出三行，第一行包含两个整数  $ans_0$ ,  $ans_1$ ，表示好的体验餐馆的起点数量和好的体验景区的起点数量。

接下来一行包含  $ans_0$  个整数  $x_i$ ，按递增顺序标识能作为起点的餐馆序列。

接下来一行包含  $ans_1$  个整数  $y_i$ ，按递增顺序标识能作为起点的景区序列。

**【样例 1 输入】**

```
4 3 8 12
0 0
1 1
2 0
3 2
2 0
3 0
0 2
2 2
0 2
1 2
2 0
1 3
2 3
1 0
1 2
1 3
1 2
1 3
1 3
0 3
4 1
0 1 2 3
0
```

**【样例 1 输出】**

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

**【样例 2 输入】**

```
3 3 6 7
```

```
0 0
1 0
2 2
0 1
0 1
2 1
0 0
1 2
2 2
2 0
1 2
0 1
1 2
2 1
0 1
1
```

**【样例 2 输出】**

```
0 0
```

**【样例 3】**

见选手目录下 *our/3.in* 与 *our/3.out*

**【数据范围与约定】**

设  $N = n + m, M = k_0 + k_1$ 。

| 测试点编号 | $N$         | $M$          |
|-------|-------------|--------------|
| 1-3   | $\leq 500$  | $\leq 1000$  |
| 4-6   | $\leq 5000$ | $\leq 10000$ |
| 7-10  | $\leq 10^6$ | $\leq 10^6$  |

# Mask

## 【题目描述】

*I'm a shape shifter*

*At Poe's masquerade*

*Hiding both face and mind*

*All free for you to draw*

在一个无穷大的二维平面上，对于所有整点  $(x, y)$ ，都与  $(x+1, y), (x, y+1)$  有一条边长为 1 的双向道路连接。此外，还有  $n$  条长度为 1 的双向小路，第  $i$  条小路连接  $(x_i, y_i)$  和  $(x_i+1, y_i+1)$ 。

最开始的你在  $(0, 0)$ ，你可能会前往  $(0, 0)$  到  $(p, q)$  之间所有点，所以你想知道所有情况的最短路距离之和。

具体的，设  $dis(x, y)$  表示  $(0, 0)$  到  $(x, y)$  的最短路距离。计算

$$\sum_{x=0}^p \sum_{y=0}^q dis(x, y)$$

## 【输入格式】

从文件 *mask.in* 中读取数据。

第一行输入三个整数  $n, p, q$ ，意义如上。

接下来  $n$  行，第  $i$  行两个整数  $x_i, y_i$  ( $0 \leq x_i < p, 0 \leq y_i < q$ )，表示第  $i$  条小路额外连接  $(x_i, y_i)$  和  $(x_i+1, y_i+1)$ ，保证对于任意  $i \neq j$  有  $x_i \neq x_j$  或  $y_i \neq y_j$ 。

## 【输出格式】

输出到文件 *mask.out* 中。

输出一个整数表示答案。

## 【样例 1 输入】

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

**【样例 1 输出】**

34

**【样例 2】**见选手目录下 *mask/2.in* 与 *mask/2.out***【样例 3】**见选手目录下 *mask/3.in* 与 *mask/3.out***【数据范围与约定】**

| 测试点编号 | $n$                   | $p, q$                   | 特殊性质               |
|-------|-----------------------|--------------------------|--------------------|
| 1-3   | $1 \leq n \leq 50000$ | $1 \leq p, q \leq 5000$  |                    |
| 4-6   | $1 \leq n \leq 5000$  | $1 \leq p, q \leq 50000$ |                    |
| 7-8   | $1 \leq n \leq 50000$ | $1 \leq p, q \leq 50000$ | 保证 $x_i, y_i$ 随机生成 |
| 9-10  | $1 \leq n \leq 10^6$  | $1 \leq p, q \leq 10^6$  |                    |

# Tale

## 【题目描述】

爱丽丝啊, 请你将这满是童真的故事握在手心里,  
 为它系上一条美丽的红丝带,  
 珍藏在你少女时代梦想荟萃的地方吧。  
 如同那来自记忆彼方中的他乡巡礼者的枯萎花环,  
 散发着迷人的神秘馨香。

在某时某刻某地有一颗  $n$  个点的树, 其中树上每个点都有点权, 点  $i$  的点权为  $w_i$ , 并且每个点都是特别的, 即所有点点权互不相同。

Alice 对这棵树感到非常好奇, 于是她提出了  $q$  个问题, 每次询问她会提出两个整数  $x_i, k_i$ , 希望知道  $\text{mex}(\{w_u | \text{dist}(u, x_i) \leq k_i \wedge 1 \leq u \leq n\})$ , ( $\wedge$  是逻辑与符号)。

其中  $\text{dist}(u, v)$  表示树上点  $u$  到  $v$  的最短路径长度。在数学中, 一个集合的 **mex** 是指不属于这个集合内的最小非负整数。

Alice 思考了很久, 但是没有想出怎么解决这个问题。但她得到了命运给她的提示: 当所有点权两两不同的时候, 这个答案要么是满足  $\text{dist}(x_i, u) > k_i$  的点  $u$  的点权, 要么是从未没在这棵树出现过的值。有了这个提示, Alice 很快就想出了做法, 并且把这题告诉你, 来考验一下你的水平。

## 【输入格式】

从文件 `tale.in` 中读取数据。

第一行包含两个正整数  $n, q$  ( $1 \leq n, q \leq 5 \times 10^5$ ), 表示节点的数量和询问数。

第二行包括  $n$  个整数  $w_1, w_2, \dots, w_n$  ( $0 \leq w_i \leq 10^9$ ), 依次表示每个点的点权。输入保证所有点的点权两两不同。

接下来的  $n - 1$  行, 每行包含三个整数  $u, v, l$  ( $1 \leq u \neq v \leq n, 1 \leq l \leq 10^9$ ), 表示树的一条连接  $u$  和  $v$  的长度为  $l$  的边, 保证输入构成一棵树。

接下来  $q$  行, 每行两个整数  $x_i, k_i$  ( $1 \leq x_i \leq n, 0 \leq k_i \leq 10^{15}$ ), 描述每个询问。

## 【输出格式】

输出到文件 `tale.out` 中。

对于每个询问, 输出一行一个整数, 即 **mex** 的值。

**【样例 1 输入】**

```

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

```

**【样例 1 输出】**

```

1
0
3
4

```

**【样例 2】**

见选手目录下 *tale/2.in* 与 *tale/2.out*

**【样例 3】**

见选手目录下 *tale/3.in* 与 *tale/3.out*

**【数据范围与约定】**

| 测试点编号 | $n$                  | $q$                  | 特殊性质              |
|-------|----------------------|----------------------|-------------------|
| 1-4   | $\leq 1000$          | $\leq 1000$          |                   |
| 5-8   | $\leq 1000$          | $\leq 5 \times 10^5$ |                   |
| 9-10  | $\leq 10^5$          | $\leq 10^5$          | 保证树是一条链           |
| 11-12 | $\leq 10^5$          | $\leq 10^5$          | 保证树是一个以 1 为中心的菊花图 |
| 13-16 | $\leq 10^5$          | $\leq 10^5$          |                   |
| 17-20 | $\leq 5 \times 10^5$ | $\leq 5 \times 10^5$ |                   |



# Box

## 【题目描述】

*I can't tell you*

*How to see me*

*Just a cage of bones*

*There's nothing inside*

在数轴上有  $n$  个箱子。每个箱子  $i$  最初放置在坐标  $x_i$ 。由于某些原因，每个箱子要被移动到新的位置  $y_i$ 。

你有一个专门的机器人帮助你完成移动箱子，机器人可以以每秒一个单位长度的速度向数轴的正负方向移动。但每个时刻机器人只能携带一个箱子。机器人拾取和放下箱子所需的时间可以忽略不计，但它需要  $C$  秒来转向。最初，您可以将机器人设置在任意位置并朝向任意方向（数轴正方向或者负方向）。但在所有箱子都运输到新的位置之后，机器人必须返回其初始位置和初始方向。箱子不需要一次性就运送到目的地，即它们可以在中途放下。此外，在此过程中，多个箱子可以共存于同一位置。

你需要制定机器人的移动策略，使移动所有箱子的总时间最小化。请计算机器人移动完所有箱子回到初始位置和方向所需的最短总时间。

## 【输入格式】

从文件 *box.in* 中读取数据。

第一行包含两个整数  $n$  和  $C$  ( $1 \leq n \leq 10^5$ ,  $0 \leq C \leq 10^9$ )，表示箱子数量和改变方向所需的时间。

接下来的  $n$  行每行包含两个整数  $x_i$  和  $y_i$  ( $1 \leq x_i, y_i \leq 10^9$ ,  $x_i \neq y_i$ )，表示方框  $i$  的起始位置和终止位置。保证所有的  $x_i$  彼此不同，所有的  $y_i$  彼此不同。

## 【输出格式】

输出到文件 *box.out* 中。

输出一个整数表示机器人运输完所有箱子回到初始位置和方向所需的最短总时间。

## 【样例 1 输入】

```
3 1
1 2
4 6
5 3
```

**【样例 1 输出】**

12

**【样例 2】**见选手目录下 *box/2.in* 与 *box/2.out***【样例 3】**见选手目录下 *box/3.in* 与 *box/3.out***【样例解释】**

样例 1：从坐标 1 开始，朝正方向前进。先拿起箱子 1。移动到坐标 2。放下箱子 1。再移动到坐标 4，拿起箱子 2。移动到坐标 6，放下箱子 2。改变方向。移动到坐标 5。拿起箱子 3。移动到坐标 3，放下箱子 3，移动到坐标 1，改变方向。总耗时 12。

样例 2：从坐标 1 开始，朝正方向前进。移动到坐标 5。拿起箱子 1。移动到坐标 8。放下箱子 1。拿起箱子 3。改变方向。移动到坐标 6。放下箱子 3。改变方向。移动到坐标 8。拿起箱子 1。移动到坐标 10。放下箱子 1。改变方向。移动到坐标 9。拿起箱子 2。移动到坐标 1。放下箱子 2。改变方向。总耗时 38。

样例 3：从坐标 3 开始，朝正方向前进。拿起箱子 3。移动到坐标 1003。放下箱子 3。移动到坐标 1004。拿起箱子 4。改变方向。移动到坐标 1002。放下箱子 4。拿起箱子 2。移动到坐标 2。放下箱子 2。移动到坐标 1。拿起箱子 1。改变方向。移动到坐标 1001。放下箱子 1。移动到坐标 1002。拿起箱子 4。改变方向。移动到坐标 4。放下箱子 4。移动到坐标 3。改变方向。总耗时 4008。

**【数据范围与约定】**

| 测试点编号 | $n$         | $C$         | 特殊性质                        |
|-------|-------------|-------------|-----------------------------|
| 1-3   | $\leq 8$    | $\leq 10^9$ |                             |
| 4-6   | $\leq 2000$ | $= 10^9$    | $1 \leq x_i, y_i \leq 10^4$ |
| 7-12  | $\leq 2000$ | $\leq 10^9$ |                             |
| 13-15 | $\leq 10^5$ | $= 0$       |                             |
| 16-20 | $\leq 10^5$ | $\leq 10^9$ |                             |