

# 第 40 届全国信息学奥林匹克竞赛

## CCF NOI2023 模拟题

### 第二试

竞赛时间：7:30–12:20

(请选手务必仔细阅读本页内容)

#### 一. 题目概况

中文题目名称	回家路线	机器人	弹跳
英文题目与子目录名	route	robot	jump
可执行文件名	route	robot	jump
输入文件名	route.in	robot.in	jump.in
输出文件名	route.out	robot.out	jump.out
每个测试点时限	1.0 秒	3.0 秒	2.0 秒
测试点数目	20	20	25
测试点是否等分	是	是	是
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	传统型	传统型	传统型
运行内存上限	512MB	512MB	128MB

#### 二. 提交源程序文件名

对于 C++ 语言	route.cpp	robot.cpp	jump.cpp
-----------	-----------	-----------	----------

#### 三. 编译选项

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

#### 四. 注意事项:

- 文件名 (程序名和输入输出文件名) 必须使用英文小写。
- 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`。
- 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 8GB。上述时限以此配置为准。
- 只提供 Linux 格式附加样例文件。
- 特别提醒: 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。

# 1. 回家路线(route)

## 【题目描述】

猫国的铁路系统中有  $n$  个站点，从  $1 \sim n$  编号。小猫准备从 1 号站点出发，乘坐列车回到猫窝所在的  $n$  号站点。它查询了能够乘坐的列车，这些列车共  $m$  班，从  $1 \sim m$  编号。小猫将在 0 时刻到达 1 号站点。对于  $i$  号列车，它将在时刻  $p_i$  从站点  $x_i$  出发，在时刻  $q_i$  直达站点  $y_i$ ，小猫只能在时刻  $p_i$  上  $i$  号列车，也只能在时刻  $q_i$  下  $i$  号列车。

小猫可以通过多次换乘到达  $n$  号站点。一次换乘是指对于两班列车，假设分别为  $u$  号与  $v$  号列车，若  $y_u = x_v$  并且  $q_u \leq p_v$ ，那么小猫可以乘坐完  $u$  号列车后在  $y_u$  号站点等待  $p_v - q_u$  个时刻，并在时刻  $p_v$  乘坐  $v$  号列车。

小猫只想回到猫窝并且减少途中的麻烦，对此它用烦躁值来衡量。

- 小猫在站点等待时将增加烦躁值，对于一次  $t$  ( $t \geq 0$ ) 个时刻的等待，烦躁值将增加  $At^2 + Bt + C$ ，其中  $A, B, C$  是给定的常数。注意：小猫登上第一班列车前，即从 0 时刻起停留在 1 号站点的那些时刻也算作一次等待。
- 若小猫最终在时刻  $z$  到达  $n$  号站点，则烦躁值将再增加  $z$ 。

形式化地说，若小猫共乘坐了  $k$  班列车，依次乘坐的列车编号可用序列  $s_1, s_2, \dots, s_k$  表示。该方案被称作一条可行的回家路线，当且仅当它满足下列两个条件：

1.  $x_{s_1} = 1, y_{s_k} = n$
2. 对于所有  $j$  ( $1 \leq j < k$ )，满足  $y_{s_j} = x_{s_{j+1}}$  且  $q_{s_j} \leq p_{s_{j+1}}$

对于该回家路线，小猫得到的烦躁值将为：

$$q_{s_k} + (A \cdot p_{s_1}^2 + B \cdot p_{s_1} + C) + \sum_{j=1}^{k-1} (A(p_{s_{j+1}} - q_{s_j})^2 + B(p_{s_{j+1}} - q_{s_j}) + C)$$

小猫想让自己的烦躁值尽量小，请你帮它求出所有可行的回家路线中，能得到的最小的烦躁值。题目保证至少存在一条可行的回家路线。

## 【输入格式】

从文件 `route.in` 中读入数据。

第一行五个整数  $n, m, A, B, C$ ，变量意义见题目描述。

接下来  $m$  行，第  $i$  行四个整数  $x_i, y_i, p_i, q_i$ ，分别表示  $i$  号列车的出发站、到达站、出发时刻与到达时刻。

## 【输出格式】

输出到文件 `route.out` 中。

输出仅一行一个整数，表示所求的答案。

**【样例 1 输入】**

```
3 4 1 5 10
1 2 3 4
1 2 5 7
1 2 6 8
2 3 9 10
```

**【样例 1 输出】**

94

**【样例 1 解释】**

共有三条可行的回家路线：

1. 依次乘坐 1, 4 号列车，得到的烦躁值为：

$$10 + (1 \times 3^2 + 5 \times 3 + 10) + (1 \times (9 - 4)^2 + 5 \times (9 - 4) + 10) = 104$$

2. 依次乘坐 2, 4 号列车，得到的烦躁值为：

$$10 + (1 \times 5^2 + 5 \times 5 + 10) + (1 \times (9 - 7)^2 + 5 \times (9 - 7) + 10) = 94$$

3. 依次乘坐 3, 4 号列车，得到的烦躁值为：

$$10 + (1 \times 6^2 + 5 \times 6 + 10) + (1 \times (9 - 8)^2 + 5 \times (9 - 8) + 10) = 102$$

第二条路线得到的烦躁值最小为 94。

**【样例 2 输入】**

```
4 3 1 2 3
1 2 2 3
2 3 5 7
3 4 7 9
```

**【样例 2 输出】**

34

**【样例 3】**

见选手目录下的 *route/route3.in* 与 *route/route3.ans*。

该样例的数据类型与最终测试点 5 ~ 8 一致。

【样例 4】

见选手目录下的 `route/route4.in` 与 `route/route4.ans`。  
该样例的数据类型与最终测试点 11 ~ 14 一致。

【样例 5】

见选手目录下的 `route/route5.in` 与 `route/route5.ans`。  
该样例的数据类型与最终测试点 18 ~ 20 一致。

【数据范围与提示】

对于所有测试点：  
 $2 \leq n \leq 10^5$  ,  $1 \leq m \leq 2 \times 10^5$   
 $0 \leq A \leq 10$  ,  $0 \leq B, C \leq 10^6$   
 $1 \leq x_i, y_i \leq n$  ,  $x_i \neq y_i$  ,  $0 \leq p_i < q_i \leq 10^3$   
每个测试点的具体限制见下表：

测试点编号	$n$	$m$	$A, B, C$ 特殊限制	其他特殊条件
1 ~ 2	$\leq 100$	$= n - 1$	无	$y_i = x_i + 1$
3 ~ 4		$\leq 100$	$A = B = C = 0$	
5 ~ 8	$\leq 2000$	$\leq 4000$	$A = B = 0$	$x_i < y_i$
9			$A = 0$	
10			无	
11 ~ 14	$\leq 10^5$	$\leq 2 \times 10^5$	$A = B = 0$	无
15			$A = 0$	
16 ~ 17			无	
18 ~ 20			无	



## 2. 机器人(robot)

### 【题目描述】

小 R 喜欢研究机器人。

最近, 小 R 新研制出了两种机器人, 分别是 P 型机器人和 Q 型机器人。现在他要测试这两种机器人的移动能力, 测试在从左到右排成一排的  $n$  个柱子上进行, 柱子用  $1 \sim n$  依次编号,  $i$  号柱子的高度为一个正整数  $h_i$ 。机器人只能在相邻柱子间移动, 即: 若机器人当前在  $i$  号柱子上, 它只能尝试移动到  $i-1$  号和  $i+1$  号柱子上。

每次测试, 小 R 会选取一个起点  $s$ , 并将两种机器人都放置在  $s$  号柱子上。随后它们会按自己的规则移动。

P 型机器人会一直向左移动, 但它无法移动到比起点  $s$  更高的柱子上。更具体地, P 型机器人在  $l$  ( $l \leq s$ ) 号柱子停止移动, 当且仅当下列两个条件均成立:

- $l = 1$  或  $h_{l-1} > h_s$ 。
- 对于满足  $l \leq j \leq s$  的  $j$ , 有  $h_j \leq h_s$ 。

Q 型机器人会一直向右移动, 但它只能移动到比起点  $s$  更低的柱子上。更具体地, Q 型机器人在  $r$  ( $r \geq s$ ) 号柱子停止移动, 当且仅当下列两个条件均成立:

- $r = n$  或  $h_{r+1} \geq h_s$ 。
- 对于满足  $s < j \leq r$  的  $j$ , 有  $h_j < h_s$ 。

现在, 小 R 可以设置每根柱子的高度,  $i$  号柱子可选择的高度范围为  $[A_i, B_i]$ , 即  $A_i \leq h_i \leq B_i$ 。小 R 希望无论测试的起点  $s$  选在哪里, 两种机器人移动过的柱子数量的差的绝对值都小于等于 2。他想知道有多少种柱子高度的设置方案满足要求, 小 R 认为两种方案不同当且仅当存在一个  $k$ , 使得两种方案中  $k$  号柱子的高度不同。请你告诉他满足要求的方案数模  $10^9 + 7$  后的结果。

### 【输入格式】

从文件 `robot.in` 中读入数据。

第一行一个正整数  $n$ , 表示柱子的数量。

接下来  $n$  行, 第  $i$  行两个正整数  $A_i, B_i$ , 分别表示  $i$  号柱子的最小和最大高度。

### 【输出格式】

输出到文件 `robot.out` 中。

仅一行一个整数, 表示答案模  $10^9 + 7$  的值。

**【样例 1 输入】**

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

**【样例 1 输出】**

```
1
```

**【样例 1 解释】**

柱子高度共两种情况：

1. 高度为：3 2 3 2 3。此时若起点设置在 5，P 型机器人将停在 1 号柱子，共移动 4 个柱子。Q 型机器人停在 5 号柱子，共移动 0 个柱子，不符合条件。
2. 高度为：3 2 4 2 3。此时无论起点选在哪，都满足条件，具体见下表：

起点编号	P 型机器人	Q 型机器人
1	停在 1 号柱子，移动过 0 个	停在 2 号柱子，移动过 1 个
2	停在 2 号柱子，移动过 0 个	停在 2 号柱子，移动过 0 个
3	停在 1 号柱子，移动过 2 个	停在 5 号柱子，移动过 2 个
4	停在 4 号柱子，移动过 0 个	停在 4 号柱子，移动过 0 个
5	停在 4 号柱子，移动过 1 个	停在 5 号柱子，移动过 0 个

【样例 2】

见选手目录下的 *robot/robot2.in* 与 *robot/robot2.ans*。

【样例 3】

见选手目录下的 *robot/robot3.in* 与 *robot/robot3.ans*。

【样例 4】

见选手目录下的 *robot/robot4.in* 与 *robot/robot4.ans*。

【数据范围与提示】

对于所有测试数据： $1 \leq n \leq 300$ ， $1 \leq A_i \leq B_i \leq 10^9$ 。  
每个测试点的具体限制见下表：

测试点编号	$n \leq$	特殊性质
1, 2	7	$A_i = B_i, B_i \leq 7$
3, 4		$B_i \leq 7$
5, 6, 7	50	$B_i \leq 100$
8, 9, 10	300	$B_i \leq 10000$
11, 12	50	$A_i = 1, B_i = 10^9$
13, 14, 15		无
16, 17	150	
18, 19	200	
20	300	



### 3. 弹跳 (jump)

#### 【题目描述】

跳蚤国有  $n$  座城市，分别编号为  $1 \sim n$ ，1 号城市为首都。所有城市分布在一个  $w \times h$  范围的网格上。每座城市都有一个整数坐标  $(x, y)$  ( $1 \leq x \leq w, 1 \leq y \leq h$ )，不同城市的坐标不相同。

在跳蚤国中共有  $m$  个弹跳装置，分别编号为  $1 \sim m$ ，其中  $i$  号弹跳装置位于  $p_i$  号城市，并具有参数  $t_i, L_i, R_i, D_i, U_i$ 。利用该弹跳装置，跳蚤可花费  $t_i$  ( $t_i > 0$ ) 个单位时间，从  $p_i$  号城市跳至坐标满足  $L_i \leq x \leq R_i, D_i \leq y \leq U_i$  ( $1 \leq L_i \leq R_i \leq w, 1 \leq D_i \leq U_i \leq h$ ) 的任意一座城市。需要注意的是，一座城市中可能存在多个弹跳装置，也可能没有弹跳装置。

由于城市间距离较远，跳蚤们必须依靠弹跳装置出行。具体来说，一次出行将经过若干座城市，依次经过的城市的编号可用序列  $a_0, a_1, \dots, a_k$  表示；在此次出行中，依次利用的弹跳装置的编号可用序列  $b_1, b_2, \dots, b_k$  表示。其中每座城市可在序列  $\{a_j\}$  中出现任意次，每个弹跳装置也可在序列  $\{b_j\}$  中出现任意次，且满足，对于每个  $j$  ( $1 \leq j \leq k$ )，编号为  $b_j$  的弹跳装置位于城市  $a_{j-1}$ ，且跳蚤能通过该弹跳装置跳至城市  $a_j$ 。我们称这是一次从城市  $a_0$  到城市  $a_k$  的出行，其进行了  $k$  次弹跳，共花费  $\sum_{i=1}^k t_{b_i}$  个单位时间。

现在跳蚤国王想知道，对于跳蚤国除首都（1 号城市）外的每座城市，从首都出发，到达该城市最少需要花费的单位时间。跳蚤国王保证，对每座城市，均存在从首都到它的出行方案。

#### 【输入格式】

从文件 `jump.in` 中读入数据。

第一行包含四个整数  $n, m, w, h$ ，变量的具体意义见题目描述。

接下来  $n$  行，第  $i$  行包含两个整数  $x_i, y_i$ ，表示  $i$  号城市的坐标。

接下来  $m$  行，第  $i$  行包含六个整数  $p_i, t_i, L_i, R_i, D_i, U_i$ ，分别表示  $i$  号弹跳装置所在的城市编号、弹跳所需的时间、可到达的矩形范围。这些整数的具体意义见题目描述。

#### 【输出格式】

输出到文件 `jump.out` 中。

输出  $n - 1$  行，第  $i$  行包含一个整数，表示从跳蚤国首都到  $i + 1$  号城市最少需要花费的单位时间。



**【样例 1 输入】**

```
5 3 5 5
1 1
3 1
4 1
2 2
3 3
1 123 1 5 1 5
1 50 1 5 1 1
3 10 2 2 2 2
```

**【样例 1 输出】**

```
50
50
60
123
```

**【样例 2】**

见选手目录下的 *jump/jump2.in* 与 *jump/jump2.ans*。

这组样例的数据范围为  $n = 10000$ ,  $m = 20000$ ,  $w = 10000$ ,  $h = 1$ 。

**【样例 3】**

见选手目录下的 *jump/jump3.in* 与 *jump/jump3.ans*。

这组样例的数据范围为  $n = 10000$ ,  $m = 20000$ ,  $w = 10000$ ,  $h = 10000$ 。

**【数据范围与提示】**

对于所有测试点和样例满足：

$1 \leq n \leq 70000$ ,  $1 \leq m \leq 150000$ ,  $1 \leq w, h \leq n$ ,  $1 \leq t_i \leq 10000$ 。

每个测试点的具体限制见下表。

测试点编号	$1 \leq n \leq$	$1 \leq m \leq$	特殊限制
1 ~ 8	100	100	无
9 ~ 13	50000	100000	每个弹跳装置恰好可达一座城市，且 $L_i = R_i, D_i = U_i$
14 ~ 18	50000	100000	$h = 1$
19 ~ 22	25000	50000	无
23 ~ 25	70000	150000	无

请注意，本题的内存限制为 128MB。