

# 2024 年计算机编程素养提升交流活动

## 拔尖班

### 实战模拟（三）

时间：2024 年 7 月 9 日 08:00 ~ 11:30

题目名称	换装	浇水	签名	电梯
题目类型	传统型	传统型	传统型	传统型
目录	outfit	water	signature	elevator
可执行文件名	outfit	water	signature	elevator
输入文件名	outfit.in	water.in	signature.in	elevator.in
输出文件名	outfit.out	water.out	signature.out	elevator.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
测试点数目	10	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	outfit.cpp	water.cpp	signature.cpp	elevator.cpp
-----------	------------	-----------	---------------	--------------

编译选项

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

#### 注意事项

- 访问 <https://jsoi.doowzs.com> 提交代码。提交系统将在北京时间 11:30 自动关闭。
- 评测使用的处理器为 Intel(R) Core(TM) i7-12700K，运行 NOI Linux 2.0 完整版，g++ 版本为 9.4.0。评测时，栈空间的大小与内存限制相同。请注意 Windows 和 Linux 之间的区别。

## 换装（outfit）

### 【题目背景】



20XX 年，OI 比赛火爆全球，JSOI 夏令营一搞就是好几个月（甚至更久）。张老师带了很多衣服来 JSOI 夏令营现场，如果每一天都穿相同的衣服，学生会觉得张老师不爱干净；如果穿了过于休闲的衣服上课，学生会觉得张老师不太靠谱。张老师想知道如何搭配夏令营每一天的穿着，才能获得最好的讲课效果。

### 【题目描述】

JSOI 夏令营总共举行  $n$  天，张老师总共带了  $m$  件不同的衣服来参加夏令营。

在夏令营开始时（开幕式视作第 0 天），张老师的讲课效果为 0，他可以穿任意一件衣服。如果张老师昨天穿的是第  $i$  件衣服，且今天穿的是第  $j$  件衣服，那么张老师今天会收获  $a_{ij}$  单位的讲课效果。需要注意的是， $a_{ij}$  可能是负数，意味着这一天张老师的讲课效果为负。

张老师想知道，在第  $n$  天时他获得的总讲课效果的最大值是多少。

### 【输入格式】

从文件 `outfit.in` 中读入数据。

第一行两个整数  $n$ 、 $m$ 。

接下来  $m$  行，每行  $m$  个整数，第  $i$  行的第  $j$  个数表示  $a_{ij}$  的值。

### 【输出格式】

输出到文件 `outfit.out` 中。

输出一个整数，表示张老师在第  $n$  天时总讲课效果的最大值。

**【样例 1 输入】**

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

**【样例 1 输出】**

```
1 5
```

**【样例 1 解释】**

张老师在第 0 天至第 2 天穿的衣服分别是：(1, 2, 1)，讲课效果为  $2 + 3 = 5$ 。

**【样例 2 输入】**

```
1 3 3
2 -9 9 9
3 -9 -1 3
4 -9 1 -1
```

**【样例 2 输出】**

```
1 13
```

**【样例 2 解释】**

张老师在第 0 天至第 3 天穿的衣服分别是：(1, 2, 3, 2)，讲课效果为  $9 + 3 + 1 = 13$ 。

**【样例 3 输入】**

```
1 114514 3
2 1 -9 1
3 -9 8 -1
4 0 -6 6
```

【样例 3 输出】

```
1 916112
```

【样例 4 输入】

```
1 12450 9
2 -9 0 0 0 1 0 0 0 0
3 -1 -9 0 0 0 0 2 0 0
4 -2 -2 -9 0 3 -6 0 0 3
5 -3 0 0 -9 0 0 -3 0 0
6 -4 7 0 0 -9 0 0 0 0
7 -5 0 -5 0 0 -9 5 8 0
8 -6 0 4 0 0 0 -9 0 4
9 -7 0 0 0 0 0 0 -9 0
10 -8 0 -7 0 2 0 0 0 -9
```

【样例 4 输出】

```
1 49803
```

【样例 5】

见选手目录下的 *outfit/outfit5.in* 与 *outfit/outfit5.ans*。

【数据规模】

对于 100% 的数据， $1 \leq n \leq 10^{15}$ ， $1 \leq m \leq 100$ ， $|a_{ij}| \leq 10$ 。

【子任务】

测试点	特殊限制
1-2	$a_{ij} = a_{ji}$
3-4	$2 \leq m \leq 3$
5-6	$1 \leq n \leq 10^3$
7-10	无特殊限制

## 浇水（water）

### 【题目背景】



氮星上有一种神奇的植物，这种植物只需要浇水就可以快速成长。

### 【题目描述】

氮星人种了一排  $n$  株神奇的植物。一开始，第  $i$  株植物的高度为  $h_i$ 。

氮星人给植物浇水的方式比较奇特。氮星人每次可以选择一段连续的区间  $[l, r]$  ( $1 \leq l \leq r \leq n$ ) 进行浇水，浇水后这段区间内的植物的高度均增加 1。

氮星人觉得这些植物长得参差不齐，不够美观，最好要让植物的高度形成“倒 V 形”。换句话说，如果植物的高度是  $H_1, H_2, \dots, H_n$ ，存在整数  $k$  ( $1 < k < n$ )，使得：

- 对于任意的  $1 \leq i < k$ ，有  $H_i < H_{i+1}$ ，且
- 对于任意的  $k \leq i < n$ ，有  $H_i > H_{i+1}$ 。

氮星人想知道，在只浇水、不修剪、不重新排列的情况下，最少需要浇水多少次，才能使  $n$  株植物的高度形成“倒 V 形”？

### 【输入格式】

从文件 `water.in` 中读入数据。

第一行一个整数  $n$ 。

第二行  $n$  个整数，表示  $h_i$  的值。

### 【输出格式】

输出到文件 `water.out` 中。

输出一个整数，表示使植物的高度形成“倒 V 形”的最少的浇水次数。

**【样例 1 输入】**

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

**【样例 1 输出】**

```
1 3
```

**【样例 1 解释】**

氮星人对区间  $[2, 2]$  浇水三次，得到结果  $H = (4, 5, 3, 1)$ 。

**【样例 2 输入】**

```
1 5
2 1 8 8 8 9
```

**【样例 2 输出】**

```
1 2
```

**【样例 2 解释】**

氮星人分别对区间  $[3, 4]$  浇水一次，再对区间  $[4, 4]$  浇水一次，得到结果  $H = (1, 8, 9, 10, 9)$ 。

**【样例 3 输入】**

```
1 5
2 8 8 8 8 8
```

**【样例 3 输出】**

```
1 2
```

【样例 4 输入】

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

【样例 4 输出】

```
1 4
```

【样例 5】

见选手目录下的 *water/water5.in* 与 *water/water5.ans*。

【数据规模】

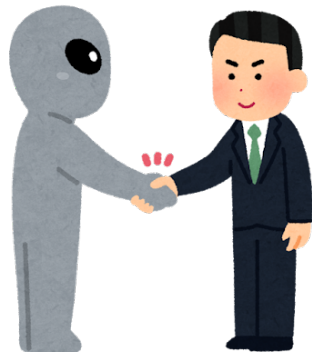
对于 60% 的数据， $3 \leq n \leq 2000$ 。  
对于 100% 的数据， $3 \leq n \leq 2 \times 10^5$ ， $1 \leq h_i \leq 10^9$ 。

【子任务】

测试点	$n$	$h_i$
1	$n \leq 2000$	单调递增
2		均相等
3-6		无特殊限制
7-10	无特殊限制	

## 签名（signature）

### 【题目背景】



氪星人费尽千辛万苦来到了 JSOI 活动现场，它能见到自己崇拜的 JSOI 选手吗？

### 【题目描述】

在周长为  $C$  的圆形湖边开办了一场 JSOI 知名选手见面会活动，活动共有  $n$  名知名选手参加。以正北方（12 点位置处）为原点，第  $i$  位选手在顺时针距离  $s_i$  的位置布置了一个摊位，并会在  $t_i$  时间收摊走人。所有选手的摊位都在不同位置，不存在两个选手位置相同的情况。

$t = 0$  时，氪星人位于 12 点位置（即顺时针 0 米处），它每隔 1 个时间单位可以顺时针或逆时针移动 1 米。如果氪星人能够在  $t_i$  时间前（含  $t_i$ ）到达选手的摊位，就可以获得选手的亲笔签名。

氪星人想知道，它最多能收集多少个不同选手的签名？

### 【输入格式】

从文件 *signature.in* 中读入数据。

第一行两个整数  $C$ 、 $n$ 。

第二行  $n$  个整数表示  $s_i$  的值。

第三行  $n$  个整数表示  $t_i$  的值。

### 【输出格式】

输出到文件 *signature.out* 中。

输出一个整数，表示氪星人能够获得不同选手的签名的最大数量。



**【样例 1 输入】**

```
1 20 4
2 5 7 10 17
3 3 1 8 2
```

**【样例 1 输出】**

```
1 0
```

**【样例 1 解释】**

氪星人无法获得任何知名选手的签名。

**【样例 2 输入】**

```
1 20 5
2 2 3 5 17 18
3 8 3 15 4 5
```

**【样例 2 输出】**

```
1 4
```

**【样例 2 解释】**

氪星人先逆时针移动 2 米，在  $t = 2$  时获得选手 5 的签名，随后再逆时针移动 1 米，在  $t = 3$  时获得选手 4 的签名。接下来，氪星人一路顺时针移动，在  $t = 8$  时获得选手 1 的签名，在  $t = 11$  时获得选手 3 的签名。

**【样例 3 输入】**

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

【样例 3 输出】

```
1 5
```

【样例 4 输入】

```
1 100 15
2 1 3 5 7 9 47 49 50 51 53 91 93 95 97 99
3 9 8 3 200 50 139 84 99 996 30 25 99 88 33 72
```

【样例 4 输出】

```
1 13
```

【样例 5】

见选手目录下的 *signature/signature5.in* 与 *signature/signature5.ans*。

【数据规模】

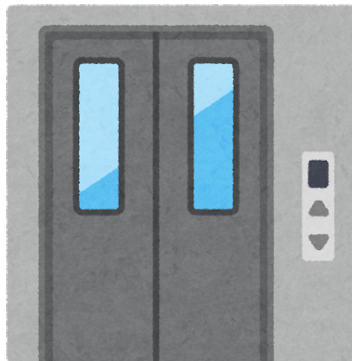
对于 100% 的数据， $5 \leq C \leq 10^9$ ， $1 \leq n \leq 200$ ， $1 \leq s_i \leq C$ ， $1 \leq t_i \leq 10^9$ 。输入保证不存在  $i \neq j$ ，使得  $s_i \neq s_j$ 。

【子任务】

测试点	$C$	$n$	$t_i$
1-2	$C \leq 100$	$n \leq 5$	$t_i \leq 100$
3-5		$n \leq 10$	
6-8	$C \leq 1000$	$n \leq 100$	$t_i \leq 1000$
9-14	无特殊限制	无特殊限制	
15-20			无特殊限制

## 电梯（elevator）

### 【题目背景】



氪星人非常喜欢玩地球上的魂类动作冒险游戏。氪星人来到了大书库，它想乘坐楼内的电梯却发现“机关不动”或是“不能从这一侧打开”，这让它十分沮丧。

### 【题目描述】

大书库是一座  $n$  层楼的复杂建筑，大书库内没有楼梯，但有  $m$  部**单向电梯**。第  $i$  部电梯从第  $a_i$  层楼前往第  $b_i$  层楼，乘坐这部电梯需要耗费  $c_i$  时间。可能存在**多个**出发和到达楼层都相同的电梯，但它们的运行耗时不同。也可能存在从某一层出发，最后又**回到同一层**的电梯，电梯关门后氪星人等待了一段时间，发现还在原地没动。

大书库入口处有  $m$  个机关。拉动第  $i$  个机关后需要在入口处等待  $d_i$  时间，此后第  $i$  个电梯的运行方向会发生反转，只能从第  $b_i$  层楼前往第  $a_i$  层楼，耗时不变。拉动某一个机关后，其他的机关都会失效不动，也就是说**最多只能拉一个机关**。氪星人可以不拉机关直接乘坐电梯；也可以拉动一个机关，改变某部电梯的运行方向后再出发。

大书库的入口位于 1 楼，顶楼存有重要道具。氪星人需要从 1 楼出发，乘坐电梯到达  $n$  楼，随后再带着重要道具乘坐电梯返回到 1 楼。氪星人想知道，假设每次乘坐电梯都不需要等待，每层楼内的移动时间均为零，它需要耗费的最短时间是多少？

### 【输入格式】

从文件 `elevator.in` 中读入数据。

第一行两个整数  $n$ 、 $m$ 。

接下来  $m$  行，每行四个整数，分别表示  $a_i$ 、 $b_i$ 、 $c_i$ 、 $d_i$ 。

### 【输出格式】

输出到文件 `elevator.out` 中。

输出一个整数，表示总费用的最小值。

**【样例 1 输入】**

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

**【样例 1 输出】**

```
1 10
```

**【样例 1 解释】**

改变第 2 部电梯（1 → 2）的运行方向后，从 1 楼出发到达 4 楼的最短路径为 1 → 2 → 4，从 4 楼出发到达 1 楼的最短路径为 4 → 3 → 1，总费用为  $1 + (3 + 3) + (1 + 2) = 10$ 。

**【样例 2 输入】**

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

**【样例 2 输出】**

```
1 12
```

**【样例 2 解释】**

不改变任何电梯的运行方向时的费用最少。

【样例 3 输入】

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

【样例 3 输出】

```
1 2
```

【样例 4】

见选手目录下的 *elevator/elevator4.in* 与 *elevator/elevator4.ans*。  
这是符合测试点 8-12 的特殊限制的样例。

【样例 5】

见选手目录下的 *elevator/elevator5.in* 与 *elevator/elevator5.ans*。

【数据规模】

对于 100% 的数据， $2 \leq n \leq 200$ ， $1 \leq m \leq 5 \times 10^4$ ， $1 \leq a_i, b_i \leq n$ ， $0 \leq c_i \leq 10^6$ ， $0 \leq d_i \leq 10^9$ 。输入保证至少存在一种满足条件的方案。

【子任务】

测试点	特殊限制
1-2	$c_i = 0$
3-7	$m \leq 1000$
8-12	$m$ 是偶数， $a_{2i-1} = a_{2i}$ ， $b_{2i-1} = b_{2i}$ ， $c_{2i-1} = c_{2i}$ ， $d_{2i-1} = d_{2i}$
13-20	无特殊限制