

# 2024 年暑期信息学奥赛考核

## 实战模拟(一)

时长：4.5 小时

题目名称	卖房	这破路也能?	异或炸弹	LIS and Inversion
题目类型	传统型	传统型	传统型	传统型
输入文件名	T1.in	T2.in	T3.in	T4.in
输出文件名	T1.out	T2.out	T3.out	T4.out
每个测试点时限	1.0 秒	1.0 秒	2.0 秒	2.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
测试点数目	20	20	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	T1.cpp	T2.cpp	T3.cpp	T4.cpp
-----------	--------	--------	--------	--------

编译选项

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

注意事项：

- 1. 本场测试只允许使用 C++ 语言。
- 2. 文件名 (程序名和输入输出文件名) 必须严格按照题目要求。
- 3. 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`。
- 4. 在机房测试的同学首先新建一个测试文件夹，文件夹名称为**学校和姓名**，并将所有的程序文件放到该文件夹内。
- 5. 若无特殊说明，题目中一行内有多个输入或输出时，默认使用空格间隔，结果的比较方式为全文比较 (即过滤行末空格及文末回车)。
- 6. 测试结束时将**文件夹**提交到教师机上。

# 卖房

【题目描述】

你是一个房地产的销售，你负责的区域可以视为一个数轴，数轴上有  $N$  个房子，编号从  $0$  到  $N - 1$ 。

现在有  $M$  个买家愿意购买房子，每个他们愿意以  $G_i$  的金额收购  $[S_i, E_i]$  范围内的所有房子。

你作为一名厉害的销售，你需要有策略地选择并销售房屋使自己的收入最大化。

需要注意的是：同一所房屋 不能 卖给不同的买家，并且允许保留一些房屋不进行出售。

【输入格式】

从文件 `T1.in` 读入数据。

输入的第一行包含两个整数  $N$  和  $M$ ，分别代表房子个数和买家个数；

接下来  $M$  行，每行包含三个整数  $S_i, E_i, G_i$ ，代表该买家愿意以  $G_i$  的金额收购  $[S_i, E_i]$  范围内的所有房子。

【输出格式】

输出到文件 `T1.out` 中。

输出一行，代表你能获得的最大收入。

【样例 1 输入】

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

【样例 1 输出】

```
3
```

【样例解释】

共有 5 所房屋，给这些房屋编号从 0 到 4，共有 3 个买家。

将位于  $[0, 0]$  范围内的房屋以 1 金币的价格出售给第 1 位买家，并将位于  $[1, 3]$  范围内的房屋以 2 金币的价格出售给第 3 位买家。

因此最多只能获得 3 枚金币。

【样例 2 输入】

```
5 3
0 0 1
0 2 10
```

1 3 2

**【样例 2 输出】**

10

**【数据范围】**

- 对于 20% 的数据：
  - $1 \leq N, M \leq 100$ ;
- 对于 60% 的数据：
  - $1 \leq N, M \leq 10^5$ ;
- 对于 100% 的数据：
  - $1 \leq N, M \leq 10^9$ ;
- $1 \leq G_i \leq 10^3$ ;
- $0 \leq S_i \leq E_i \leq N - 1$ ;
- 所有输入值都是整数。

# 这破路也能？

【题目描述】

现代社会，路是必不可少的。

共有  $N$  个城镇， $M$  条道路，任意两个城镇都有路相连，而且往往不止一条。

但有些路年久失修，走着很不爽。

按理说条条大路通罗马，大不了绕行其他路呗——可你却发现：从  $A$  城到  $B$  城不管怎么走，总有一些逃不掉的必经之路。

现在请你计算一下， $A$  到  $B$  的所有路径中，有几条路是逃不掉的？

【输入格式】

从文件 `T2.in` 读入数据。

输入的第一行包含两个整数  $N$  和  $M$ ，分别表示城镇数和道路数；

接下来  $M$  行，每行两个整数  $x$  和  $y$ ，表示  $x$  城和  $y$  城之间有一条长为 1 的双向路；

第  $M + 2$  行包含一个整数  $Q$ ，代表询问次数；

接下来  $Q$  行，每行两个整数  $A$  和  $B$ ，表示一次询问。

【输出格式】

输出到文件 `T2.out` 中。

对于每次询问，输出一个正整数，表示  $A$  城到  $B$  城必须经过几条路。

每个输出占一行。

【样例 1 输入】

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

【样例 1 输出】

```
0
1
```

**【数据范围】**

- 对于 20% 的数据：
  - $0 \leq N \leq 100$ ;
  - $0 \leq M \leq 200$ ;
  - $0 \leq Q \leq 50$ ;
- 对于 70% 的数据：
  - $0 \leq N \leq 10^4$ ;
  - $0 \leq M \leq 2 \times 10^4$ ;
  - $0 \leq Q \leq 10^4$ ;
- 对于 100% 的数据：
  - $0 \leq N \leq 10^5$ ;
  - $0 \leq M \leq 2 \times 10^5$ ;
  - $0 \leq Q \leq 10^5$ ;
- $1 \leq x, y, A, B \leq N$ ;
- 对于任意的道路，两端两端的城市编号之差不超过  $10^4$
- 任意两个城镇都有道路相连，同一条道路不会出现两次；
- 道路的起终点不会相同，查询的两个城市不会相同；
- 所有输入值都是整数。

# 异或炸弹

【题目描述】

给定一个  $N \times N$  的矩阵，初始全是 0，现在你手上有  $m$  个炸弹，对于每一个炸弹，都有自己的爆炸中心  $(x,y)$  和爆炸半径  $r$ 。

当矩阵内某个位置与爆炸中心的曼哈顿距离小于等于  $r$  时，该位置就会收到爆炸的影响，爆炸的影响就是给这个位置上的数异或 1。

给你这  $m$  个炸弹的爆炸位置和爆炸半径，你需要回答这个矩阵中 1 的个数。

曼哈顿距离：两个点  $(x_1,y_1)$  和  $(x_2,y_2)$  的曼哈顿距离为  $d = |x_1 - x_2| + |y_1 - y_2|$ 。

【输入格式】

从文件 `T3.in` 读入数据。

输入的第一行包含两个整数  $N$  和  $M$ ，分别表示矩阵大小和炸弹数量；

接下来  $M$  行，每行包含三个整数  $x_i,y_i$  和  $r_i$ ，表示第  $i$  个炸弹的爆炸中心的坐标是  $(x_i,y_i)$ ，爆炸半径是  $r_i$ 。

【输出格式】

输出到文件 `T3.out` 中。

输出被轰炸后的矩阵中 1 的个数。

【样例 1 输入】

```
5 1
3 3 1
```

【样例 1 输出】

```
5
```

【样例 1 解释】

```
0 0 0 0 0
0 0 1 0 0
0 1 1 1 0
0 0 1 0 0
0 0 0 0 0
```

爆炸结果如上所示，共有 5 个 1。

【数据范围】

- 对于 20% 的数据：

- $1 \leq N \leq 100$ ;
- $1 \leq M \leq 100$ ;
- 对于 70% 的数据：
  - $1 \leq N \leq 2000$ ;
  - $1 \leq M \leq 6000$ ;
- 对于 100% 的数据：
  - $1 \leq N \leq 3000$ ;
  - $1 \leq M \leq 5 \times 10^5$ ;
- $1 \leq x_i, y_i \leq N$ ;
- $0 \leq r_i \leq 6000$ ;
- 所有输入值都是整数。

# LIS and Inversion

【题目描述】

给你一个长度为  $N$  的序列  $A$ ，满足  $0 \leq A_i < i$ 。  
定义一个排列  $P$  的得分为它的最长上升子序列长度，同时定义其代价为满足以下条件的正整数  $i$  的数量：

- 只存在小于  $A_i$  个位置的  $j < i$ ，使得  $P_j > P_i$ 。

对每个  $k = 1, 2, \dots, n$ ，求所有得分不小于  $k$  的排列的最小代价。

【输入格式】

从文件 `T4.in` 读入数据。  
输入的第一行包含一个整数  $N$ ；  
输入的第二行包含  $N$  个整数  $A_1, A_2, \dots, A_N$ ；

【输出格式】

输出到文件 `T4.out` 中。  
按  $k = 1, 2, \dots, N$  的顺序输出结果。

【样例 1 输入】

```
4
0 1 2 1
```

【样例 1 输出】

```
0 0 1 3
```

【样例 1 解释】

- 对于各个  $k$  值，解  $P$  如下所示：
- $k = 1$ ：当  $P = (4, 2, 1, 3)$  时， $P$  的得分为 2，成本为 0。
  - $k = 2$ ：当  $P = (4, 3, 1, 2)$  时， $P$  的得分为 2，成本为 0。
  - $k = 3$ ：当  $P = (4, 1, 2, 3)$  时， $P$  的得分为 3，成本为 1。
  - $k = 4$ ：当  $P = (1, 2, 3, 4)$  时， $P$  的得分为 4，成本为 3。

【样例 2 输入】

```
3
0 0 0
```

【样例 2 输出】

```
0 0 0
```



**【样例 3 输入】**

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

**【样例 3 输出】**

```
0 1 2 3 4
```

**【样例 4 输入】**

```
11
0 0 2 3 4 5 3 7 8 2 10
```

**【样例 4 输出】**

```
0 0 0 1 2 3 4 5 7 8 9
```

**【数据范围】**

- 对于 20% 的数据：
  - $1 \leq N \leq 100$ ;
- 对于 50% 的数据：
  - $1 \leq N \leq 1000$ ;
- 对于 100% 的数据：
  - $1 \leq N \leq 2.5 \times 10^5$ ;
- $0 \leq A_i < i$ ;
- 所有输入值都是整数。