

# “科大国创杯” 2021 年安徽省青少年信息学奥林匹克竞赛

## 初中组试题

### AOI 2021

比赛时间：2021 年 4 月 10 日 14:00–18:00

|       |             |          |          |            |
|-------|-------------|----------|----------|------------|
| 题目名称  | 运气游戏        | 坑        | 收衣服      | 地铁         |
| 题目类型  | 传统型         | 传统型      | 传统型      | 传统型        |
| 题目英文名 | mahjong     | hole     | sort     | subway     |
| 输入文件名 | mahjong.in  | hole.in  | sort.in  | subway.in  |
| 输出文件名 | mahjong.out | hole.out | sort.out | subway.out |
| 时间限制  | 1.0 秒       | 1.0 秒    | 1.0 秒    | 1.0 秒      |
| 内存限制  | 512 MB      | 512 MB   | 512 MB   | 512 MB     |
| 测试点数目 | 20          | 20       | 20       | 20         |

提交源程序文件名

|           |             |          |          |            |
|-----------|-------------|----------|----------|------------|
| C++ 语言    | mahjong.cpp | hole.cpp | sort.cpp | subway.cpp |
| C 语言      | mahjong.c   | hole.c   | sort.c   | subway.c   |
| Pascal 语言 | mahjong.pas | hole.pas | sort.pas | subway.pas |

编译选项

|           |     |
|-----------|-----|
| C++ 语言    | -lm |
| C 语言      | -lm |
| Pascal 语言 |     |

**注意事项（请选手仔细阅读）：**

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. 选手需要在桌面上建立以选手的参赛号为名的目录，并由选手为每道题再单独建立一个子目录，子目录名与对应的试题英文名相同。选手提交的每道试题的源程序必须存放在相应的子目录下。
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。

## 1 运气游戏 (mahjong)

### 【题目背景】

你可以选择跳过背景部分。

初春的一天，正是乍暖还寒时候，狂风乍起。小可可裹紧了单薄的外衣，往小雪家中赶去。

“今天真不是个出门的时候啊！”小可可感叹道。

小雪附和了一句。过了一会儿，除了几句客套外仿佛也聊不出什么话题了。

突然，小可可想到了一个绝妙的打发时间的办法：麻将！

“就算你这么说，可我不会打麻将啊！”

“你打开软件跟着学一下，再打打，不久就会了。”

小雪很快就加入了一局麻将，是庄家，也就是第一个出牌的人。奇怪的是，小雪刚摸到第一张牌，系统就显示了一个“和”的按钮。

“怎么可能？系统出问题了吧？”

“这也太简单了吧！麻将只不过是个运气游戏吧？”

不服气的小可可仔细看了看小雪的手牌，发现居然确实和了，这种一张牌没出就取胜的罕见现象叫作天和。更稀奇的是，小雪的牌型实际上还符合了另一种少见的役——四暗刻。

### 【问题描述】

如果每个人都有小雪一样的好运气的話，麻将就变成一个规则很简单的游戏了——你只需要验证你的手牌有没有和成四暗刻。

麻将一共有 34 种牌，分为 5 类——万子、筒子、索子、风牌和三元牌：

1. 万字、筒子和索子各包含九种分别写有一到九的麻将牌；
2. 风牌有东风、南风、西风、北风四种；
3. 三元牌分为白、发、中三种。

以上每种牌有 4 张，总共 136 张。每个玩家手中常有 13 张牌，每回合摸一张牌形成 14 张牌，再打一张牌。

介绍四暗刻的和牌条件前，先介绍刻子和对子：

1. 刻子指的是完全相同的 3 张牌。
2. 对子指的是完全相同的 2 张牌。

当你的手牌形成了 4 个刻子和 1 个对子时，就和成四暗刻啦！

现在，小可可给了你 14 张牌，你能判断出有没有和出四暗刻呢？

### 【输入格式】

输入文件名为 mahjong.in。

输入包含多组测试数据。

第一行是一个正整数  $T$ ，表示测试数据的组数。

下面  $T$  组测试数据。每组测试数据包含一行 14 个用空格隔开的字符串，按以下的格式输入麻将牌：

1. 万子、筒子和索子用同样的格式 AB 输入，其中 A 是牌上写的数字，B 是单个字母表示牌的类型，我们用 m 表示万子，s 表示索子，p 表示筒子。例如，字符串 4m 表示四万。
2. east 表示东风，south 表示南风，west 表示西风，north 表示北风。
3. white 表示白，green 表示发，red 表示中。

输入保证每种牌的数量不超过 4。

### 【输出格式】

输出文件名为 mahjong.out。

输出  $T$  行，每行一个字符串，Yes 表示和了，No 表示没和。

注意：请选手特别注意正确的大小写。

### 【样例输入】

```
4
1p 1p 1p 2p 2p 3p 4p 5p 6p 7p 8p 9p 9p
6p white white 2s 2s 2s 3s 3s 6p 3s 4s 4s 4s 6p
1m 1m 1m 1m 2m 2m 2m 2m 3m 3m 3m 3m east east
red red red white white white green green green west west
```

### 【样例输出】

```
No
Yes
No
Yes
```

### 【样例解释】

第一组样例的牌型也很罕有，叫做九莲宝灯，但是并不是四暗刻，因为只有两组刻子。

第二组样例中的对子为两张白，四组刻子为 222s, 333s, 444s 和 666p，因此是四暗刻。

第三组样例只有三组刻子 111m, 222m, 333m，缺少一组刻子。

第四组样例是完美的四暗刻。另外，它还同时符合大三元和字一色两个珍稀的役。

### 【数据范围与提示】

- 对于 20% 的测试点， $T = 1$ ；
- 对于另外 30% 的测试点，保证牌的种类只有万子；
- 对于另外 30% 的测试点，保证牌的种类只有万字、筒子和索子；
- 对于 100% 的测试点，保证  $1 \leq T \leq 100$ 。

## 2 坑 (hole)

### 【题目背景】

你可以选择跳过背景部分。

随手天和后，麻将让小雪也提不起什么兴趣了。恶劣的天气、压抑的氛围让小雪心情越来越差，之后倒起了苦水：

“唉！今天又被一个不靠谱的同学坑了，浪费了我好多时间。”

“期中考试还早，有什么好焦虑的呢？别卷了，正好来看看最近在蛐蛐国流行的一个游戏吧。”小雪看了游戏来了精神：看起来好像很解压。

### 【问题描述】

游戏在一个左右无限延伸的数轴上进行，上面有  $n$  只跳蚤和  $m$  个坑，它们都可以被抽象成数轴上的一个点。

玩家每回合需要选择让所有跳蚤一起向左/向右跳一个单位长度。如果一个代表跳蚤的点与一个代表坑的点重合了，跳蚤就会掉进坑中，发出惨叫后死去。

郁闷的小雪想用最快的时间杀死所有跳蚤，请你帮小雪计算一下这个最少的回合数。

### 【输入格式】

输入文件名为 `hole.in`。

第一行两个空格隔开的正整数  $n, m$ 。

第二行  $n$  个空格隔开的整数  $x_1, x_2, \dots, x_n$ ，其中  $x_i$  表示第  $i$  只跳蚤初始时的坐标。

第三行  $m$  个空格隔开的整数  $y_1, y_2, \dots, y_m$ ，其中  $y_i$  表示第  $i$  个坑的坐标。

输入数据保证以上  $n + m$  个坐标两两互不相等。

### 【输出格式】

输出文件名为 `hole.out`。

仅一行一个整数，表示杀死所有跳蚤的最少回合数。

### 【样例 1 输入】

```
3 2
3 -1 2
0 10
```

### 【样例 1 输出】

```
5
```

### 【样例 1 解释】

第一回合让所有跳蚤向右跳一步，第 2 个跳蚤进第一个坑，剩下两个跳蚤分别位于 4, 3。

下面四回合让所有跳蚤向左跳，两个跳蚤都进入第一个坑，游戏结束。

**【样例 2】**

见下发文件的 hole/hole2.in 和 hole/hole2.ans。

**【数据范围与提示】**

- 对于 20% 的数据，保证  $1 \leq n \leq 20$ ；
- 对于另外 20% 的数据，保证  $1 \leq n, m \leq 300$ ；
- 对于另外 20% 的数据，保证  $1 \leq x_i, y_i \leq 2000$ ；
- 对于另外 10% 的数据，保证  $1 \leq n, m \leq 2000$ ；
- 对于另外 10% 的数据，保证  $m = 2$ ；
- 对于 100% 的数据，保证  $1 \leq n, m \leq 2 \times 10^5, -10^9 \leq x_i, y_i \leq 10^9$ 。

### 3 收衣服 (sort)

#### 【题目背景】

你可以选择跳过背景部分。

沉迷于虐待跳蚤游戏的小雪没有发觉时间过了多久，一抬头发现竟然天色大变！天空一片昏黄，一股怪味扑鼻而来。没想到在如此发达的 2077 年，城市中还能碰到沙尘暴，这超现实的场景让小雪怀疑是跳蚤国王显灵。

“别愣着了，快去收衣服呀！”小可可突然想到。

#### 【问题描述】

看着这么多蒙灰的衣服，他们俩欲哭无泪；而且有的衣服是没法一起洗的，为了分门别类，小可可给了每种衣服一个  $1 \sim n$  的两两不同的标号，其中  $n$  是衣服的件数，把衣服排成  $1, 2, \dots, n$  的顺序再收会比较方便。

小可可还想到，我们可以把一段连续的晾衣架拿出来，在手上翻转顺序，再放回去。作为 OI 选手的你，马上抽象出了小可可排序衣服的算法：我们设初始时从左往右第  $i$  件衣服的标号为  $p_i$ ，按  $1, 2, \dots, n-1$  的顺序枚举  $i$ ，设  $p_i, p_{i+1}, \dots, p_n$  中标号最小的是  $p_j$ ，那么将  $p_i, p_{i+1}, \dots, p_{j-1}, p_j$  左右翻转变成  $p_j, p_{j-1}, \dots, p_{i+1}, p_i$ 。

小雪很快发现，小可可的算法看似厉害，实际很蠢——在天色的影响下，大家都分不出衣服的标号了。于是他们只能回到房间进行理性愉悦：我们假设左右翻转区间  $[i, j]$  的操作代价是  $w_{i,j}$ ，一次排序的代价是每次翻转的操作代价和。现在小可可想知道，当  $p$  取遍  $n!$  种排列时，所有情况的排序代价之和。

只用输出答案对 998244353 ( $= 7 \times 17 \times 2^{23} + 1$ ，一个质数) 取模后的值。

#### 【输入格式】

输入文件名为 `sort.in`。

第一行一个整数  $n$ 。

下面  $n-1$  行，第  $i$  ( $1 \leq i \leq n$ ) 行  $n-i+1$  个空格隔开的整数，第  $j$  个表示  $w_{i,j}$ 。

#### 【输出格式】

输入文件名为 `sort.out`。

一行一个整数表示答案。

#### 【样例 1 输入】

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

**【样例 1 输出】**

144

**【样例 2】**

见下发文件的 sort/sort2.in 和 sort/sort2.ans。

**【数据范围与提示】**

- 对于 25% 的数据，保证  $1 \leq n \leq 9$ ；
- 对于 50% 的数据，保证  $1 \leq n \leq 16$ ；
- 对于 70% 的数据，保证  $1 \leq n \leq 50$ ；
- 对于另外 15% 的数据，保证  $w_{i,j} = 1$ ；
- 对于 100% 的数据，保证  $1 \leq n \leq 500, 0 \leq w_{i,j} < 998244353$ 。

## 4 地铁

### 【题目背景】

你可以选择跳过背景部分。

小可可发现自己所学算法在生活中其实无大用，感觉十分沮丧。小雪见状还是嘀咕了几句“应该还是有用的吧”。

“不过没用又怎么样呢？算法只不过是一块名牌大学的敲门砖罢了。”

“你这话我就不同意了。跳蚤国王曾经和我说过，以后科研或者工作中我们还会和信息学竞赛中的某些东西重逢，虽然可能不会再有信息学竞赛这么难。

“除开功利的因素之外，搞信息学竞赛还是能享受到很多思考的乐趣的。”

“你说的也对。每次我在考场上不会做质疑这题是不是有问题的时候，考后看题解总是懊恼又快乐——这么自然的思路我怎么想不到呢！”

一颗理论计算机科学家的种子悄悄萌芽。

沙尘暴突然神奇般的散去了。实在坐不下去的两人决定出门坐地铁瞎逛，随性下车。即使没有刻意为之，小雪在地铁上却想出了一个有意思的问题，你能解决吗？

### 【问题描述】

B 市的地铁历史悠久，小雪和小可可乘坐的 X 号线是环形路线，上面分布着  $n$  个车站，相邻两个车站之间的铁路长度为正整数。现在小雪进行了一些观察，得到了  $m$  条信息，第  $i$  条信息是如下形式之一：

1. 环上顺时针由  $S_i$  到  $T_i$  的一段距离不小于一个给定的值  $L_i$  ( $S_i$  和  $T_i$  是两个车站)；
2. 环上顺时针由  $S_i$  到  $T_i$  的一段距离不大于一个给定的值  $L_i$ 。

小雪想要你计算最后 X 线地铁的总长度有多少种不同的合法取值。

### 【输入格式】

输入文件名为 `railway.in`。

第一行两个空格隔开的正整数  $n$  和  $m$ 。

下面  $m$  行，第  $i$  行四个空格隔开的正整数  $type_i, S_i, T_i, L_i$ ，其中  $type_i \in \{1, 2\}$  表示信息的类型。车站顺时针编号为从 1 开始的连续整数。保证  $1 \leq S_i, T_i \leq n$  且  $S_i \neq T_i$ 。

### 【输出格式】

输出文件名为 `railway.out`。

仅一行一个整数，表示所求答案。如果有无穷种取值，请输出 -1。



**【样例 1 输入】**

```
4 8
1 1 3 3
2 2 4 5
1 2 4 4
1 3 1 4
2 4 2 5
1 4 2 3
```

**【样例 1 输出】**

```
4
```

**【样例 1 解释】**

定义数组  $d[1..4]$ ，其中  $d[i]$  表示  $i$  号车站顺时针到  $i+1$  号车站的铁路长度。

1.  $d = [1, 2, 2, 2]$ ，总长度为 7；
2.  $d = [1, 2, 2, 3]$ ，总长度为 8；
3.  $d = [1, 2, 2, 4]$ ，总长度为 9；
4.  $d = [1, 2, 3, 4]$ ，总长度为 10。

可以证明，不存在其他的可能长度，于是答案为 4。

**【样例 2 输入】**

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

**【样例 2 输出】**

```
-1
```

**【样例 2 解释】**

3 号车站顺时针到 1 号车站的铁路长度可以为任意正整数。

**【样例 3】**

见下发文件的 `railway/railway3.in` 和 `railway/railway3.ans`。

**【数据范围与提示】**

- 对于 30% 的数据，保证  $n, m \leq 9, L \leq 5$ ;
- 对于另外 15% 的数据，保证  $T$  是  $S$  顺时针方向后第一个车站;
- 对于另外 20% 的数据，保证  $T$  是  $S$  顺时针方向后第二个车站;
- 对于另外 25% 的数据，保证  $n, m \leq 50$ ;
- 对于 100% 的数据，保证  $3 \leq n \leq 500, 1 \leq m \leq 500, 1 \leq L_i \leq 10^9$ 。