

# 洛谷网校 CSP-S 模拟赛

## 第一试

时间：2023 年 10 月 3 日 14:00 ~ 18:00

题目名称	蠕虫病毒	定向越野	疯狂星期四	快即慢
题目类型	传统型	传统型	传统型	传统型
目录	virus	orient	thursday	fastslow
可执行文件名	virus	orient	thursday	fastslow
输入文件名	virus.in	orient.in	thursday.in	fastslow.in
输出文件名	virus.out	orient.out	thursday.out	fastslow.out
每个测试点时限	1.0 秒	2.0 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	25	10
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	virus.cpp	orient.cpp	thursday.cpp	fastslow.cpp
-----------	-----------	------------	--------------	--------------

编译选项

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

### 注意事项

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++ 中 `main` 函数的返回值类型必须是 `int`，程序正常结束时的返回值必须为 0。
3. 选手源程序必须存放于题目指定的目录（子文件夹）下。
4. 程序可使用的栈空间大小与该题内存空间限制一致。
5. 若无特殊说明，每道题的源代码大小限制为 100 KiB，输入与输出中同一行的相邻整数、字符串等均使用一个空格分隔，结果比较方式为全文比较（忽略行末空格、文末回车）。
6. 统一评测时采用的机器配置为：Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz，上述时限以此配置为准。
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此为准。
9. 选手不得在源程序中使用内嵌汇编，也禁止以任何方式修改程序的编译选项。

## 蠕虫病毒 (virus)

### 【题目描述】

沃若决定给扶苏一点小小的蠕虫病毒震撼！

具体来说，沃若手里有  $n$  种蠕虫病毒，而扶苏的电脑中有  $m$  张硬盘。第  $i$  种蠕虫病毒可以让扶苏电脑中的第  $l_i$  到第  $r_i$  张硬盘均被攻击  $k_i$  次。

现在，沃若将所有  $n$  种病毒都通过粉兔网络传输到了扶苏的电脑中，即将发动攻击。就在这时，扶苏告诉沃若自己的电脑中存放着许多洛谷机密。为了大局着想，沃若决定放下私人恩怨，保护洛谷谷民。

但是病毒已经发出，即便是沃若也不能完全撤回攻击了。沃若只能选择恰好 1 种蠕虫病毒让其失效自毁，而其余的蠕虫病毒仍然会攻击指定的硬盘。

对于每种蠕虫病毒，沃若想知道如果让其失效自毁，所有硬盘被攻击的最大次数。

### 【输入格式】

从文件 *virus.in* 中读入数据。

第一行为两个正整数  $n$  和  $m$ ，表示蠕虫病毒的数量及硬盘的数量。

随后  $n$  行，第  $i$  行有三个正整数  $l_i, r_i, k_i$ ，描述了第  $i$  种蠕虫病毒的效果。

### 【输出格式】

输出到文件 *virus.out* 中。

输出  $n$  行，每行一个整数表示答案。

### 【样例 1 输入】

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

### 【样例 1 输出】

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

**【样例 2】**

见选手目录下的 *virus/virus2.in* 与 *virus/virus2.ans*。

**【子任务】**

对于 20% 的测试数据，有  $1 \leq n, m \leq 10$ 。

对于 40% 的测试数据，有  $1 \leq n, m \leq 5,000$ 。

对于 60% 的测试数据，有  $1 \leq n, m \leq 10^5$ 。

对于 100% 的测试数据，保证  $1 \leq n, m \leq 10^6$ ， $1 \leq l_i \leq r_i \leq m$ ， $1 \leq k_i \leq 1,000$ 。

## 定向越野 (orient)

### 【题目描述】

须弥洛谷中有一棵巨大的树，称为世界树扶桑。它的形态可以用一张有  $n$  个点和  $n - 1$  条边的无向连通图表示，点的编号为 1 到  $n$ 。

现在，大家要举办一场定向越野比赛。在设计路线时，扶苏会依次选择三个点  $x, y, z (1 \leq x, y, z \leq n)$ ， $x$  为起点， $y$  为检查站， $z$  为终点。选手需要沿着  $x \rightarrow y \rightarrow z$  的简单路径完成比赛。

但沃若很快就发现这样是不可行的，因为一条合理的路线必须满足  $x \neq y, x \neq z, y \neq z$ ，且同一条边最多只会被路线覆盖一次。

请计算扶苏所有可能设计的路线中，不合理路线的数量。

### 【输入格式】

从文件 *orient.in* 中读入数据。

第一行为一个正整数  $n$ ，表示树的点数。

随后  $n - 1$  行，每行两个正整数  $u_i, v_i$ ，表示树上有一条边直接连接  $u_i$  与  $v_i$ 。

### 【输出格式】

输出到文件 *orient.out* 中。

输出一行一个整数表示答案。

### 【样例 1 输入】

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

### 【样例 1 输出】

```
1 25
```

### 【样例 2 输入】

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

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

**【样例 2 输出】**

```
1 109
```

**【样例 3】**

见选手目录下的 *orient/orient3.in* 与 *orient/orient3.ans*。

**【子任务】**

对于 10% 的测试数据，有  $2 \leq n \leq 10$ 。

对于 30% 的测试数据，有  $2 \leq n \leq 50$ 。

对于 60% 的测试数据，有  $2 \leq n \leq 5,000$ 。

对于 100% 的测试数据，保证  $2 \leq n \leq 10^6$ ， $1 \leq u_i, v_i \leq n$ 。

对于 20% 的测试数据，满足输入的树形态为一条链。

对于另外 20% 的测试数据，满足输入的树形态有  $n - 1$  片叶子。

保证输入的树合法。

## 疯狂星期四 (thursday)

### 【题目描述】

又到了疯狂星期四！沃若和扶苏来到了肯德基，买了许多蛋挞。

沃若觉得简单地吃太没意思了，拉着扶苏来玩一个吃蛋挞游戏：

- 他们都有一个得分序列，初始为空
- 第 1 轮吃的人会吃掉 1 个蛋挞，随后每轮吃的人都要比上轮多吃 1 个蛋挞
- 双方可以随时抢吃，但只有一个人吃完了本轮才能进行下一轮，同一个人可以连续多轮抢吃
- 每当一个人吃完了本轮的蛋挞，就将本轮吃的蛋挞数量加入自己的得分序列末尾
- 游戏可以在任何一轮吃完后结束，但至少进行 1 轮

---

例如，一种可能的最终得分序列是：

- 沃若：[1, 3, 4]
- 扶苏：[2, 5]

在这个例子中，沃若抢吃了第 1 轮，随后扶苏抢吃了第 2 轮，第 3, 4 轮均被沃若抢得，最终扶苏又抢吃第 5 轮。

---

已知沃若和扶苏分别可以吃下不超过  $n, m$  个蛋挞，请问最终可能的得分序列有多少种情况。两种情况视为不同，当且仅当沃若的得分序列不同或扶苏的得分序列不同。

由于答案可能很大，你只需要输出答案对  $10^9 + 7$  取模后的值。

### 【输入格式】

从文件 *thursday.in* 中读入数据。

输入由多组数据构成。

第一行一个正整数  $T$ ，表示共有  $T$  组数据。

对于每组数据，有一行两个空格分隔的正整数  $n, m$ 。

### 【输出格式】

输出到文件 *thursday.out* 中。

对于每组数据，依次输出一行一个整数表示答案。

**【样例 1 输入】**

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

**【样例 1 输出】**

```
1 28
2 2
3 10
4 14
```

**【样例 2】**

见选手目录下的 *thursday/thursday2.in* 与 *thursday/thursday2.ans*。

**【子任务】**

对于 8% 的测试数据，有  $T = 1$ ， $1 \leq n, m \leq 6$ 。

对于 16% 的测试数据，有  $T = 1$ ， $1 \leq n, m \leq 100$ 。

对于 32% 的测试数据，有  $T = 1$ ， $1 \leq n, m \leq 1,000$ 。

对于 64% 的测试数据，有  $T = 1$ ， $1 \leq n, m \leq 5,000$ 。

对于 88% 的测试数据，有  $T = 1$ ， $1 \leq n, m \leq 50,000$ 。

对于 100% 的测试数据，保证  $1 \leq T \leq 10,000$ ， $1 \leq n, m \leq 50,000$ 。

## 快即慢 (fastslow)

### 【题目描述】

给定一个长度为  $n$  的数列  $a$ 。

有  $m$  次询问，每次给出  $x$  和  $y$ 。你要把数列划分成非空的两段（一个非空前缀和一个非空后缀），使得『 $x$  在前缀中出现的次数  $\times y$  在后缀中出现的次数』最大。

形式化地，定义：

- $A(p, x)$ :  $[1, p]$  这个前缀里  $x$  的出现次数。
- $B(p, x)$ :  $[1, p]$  这个后缀里  $x$  的出现次数。

对于一组询问，你要找到一个  $p \in [2, n]$ ，最大化  $A(p-1, x) \times B(p, y)$ 。

### 【输入格式】

从文件 *fastslow.in* 中读入数据。

第一行是两个整数，表示数列长度  $n$  和询问次数  $m$ 。

第二行有  $n$  个整数，依次表示  $a_1, a_2, \dots, a_n$ 。

接下来  $m$  行，每行两个整数  $x, y$  表示一次询问。

### 【输出格式】

输出到文件 *fastslow.out* 中。

对每组询问，输出一行一个整数表示  $A(p-1, x) \times B(p, y)$  的最大值。

### 【样例 1 输入】

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

### 【样例 1 输出】

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



**【样例 2】**

见选手目录下的 *fastslow/fastslow2.in* 与 *fastslow/fastslow2.ans*。

**【子任务】**

- 对 20% 的数据,  $n, m \leq 100$ 。
- 对 40% 的数据,  $n, m \leq 1,000$ 。
- 另有 20% 的数据, 数列中每个数字的出现次数不超过 100。
- 另有 20% 的数据, 数列中的不同数字数量不超过 100。
- 对 100% 的数据,  $2 \leq n \leq 10^5$ ,  $1 \leq m \leq 10^5$ ,  $1 \leq a_i, x, y \leq 10^9$ 。