# Public NOIP Round #4

## 提高组

**时间**：2022 **年** 11 **月** 20 **日** 8:30 ~ 13:00

| 题目名称 | 治病 | 拓扑序计数 | 序列 | 水果 |
|---|---|---|---|---|
| 题目类型 | 传统型 | 传统型 | 传统型 | 传统型 |
| 可执行文件名 | `doctor` | `topo` | `loose` | `fruit` |
| 每个测试点时限 | 3.0 秒 | 2.0 秒 | 2.0 秒 | 3.0 秒 |
| 内存限制 | 512 MiB | 512 MiB | 256 MiB | 1 GiB |
| 子任务数目 | 5 | 13 | 4 | 6 |
| 测试点是否等分 | 是 | 否 | 否 | 否 |

提交源程序文件名

| 对于 C++ 语言 | `doctor.cpp` | `topo.cpp` | `loose.cpp` | `fruit.cpp` |
|---|---|---|---|---|

编译选项

| 对于 C++ 语言 | `-lm -O2` |
|---|---|

**注意事项：**

1. C++ 中函数 main() 的返回值类型必须是 int，值必须为 0。
2. 对于因未遵守以上规则对成绩造成的影响，相关申诉不予受理。
3. 若无特殊说明，输入文件中同一行内的多个整数、浮点数、字符串等均使用一个空格进行分隔。
4. 若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。
5. 程序可使用的栈空间大小与该题内存空间限制一致。
6. 在终端下可使用命令 `ulimit -s unlimited` 将栈空间限制放大，但你使用的栈空间大小不应超过题目限制。

# 治病（doctor）

## 【题目描述】

虱子国王尼特这天有点不舒服，它周围的 $n$ 个医生立刻开出了药方：第 $i$ 个医生告诉它，从这天起的第 $L_i$ 天到第 $R_i$ 天，它应该服用 $x_{i,1}, x_{i,2}, \ldots, x_{i,K_i}$ 这 $K_i$ 种药，每天每种药应当服用恰好一片。注意，如果有多个医生的药方里都要求尼特在第 $p$ 天服用第 $q$ 种药，那尼特在第 $p$ 天仍然只会服用一片第 $q$ 种药。编号为 $j$ 的药每片需要 $c_j$ 元钱。

然而，由于尼特的疏忽，有恰好一位庸医混进了医生队伍里，但尼特并不知道哪位医生是庸医。所以它想知道，对于所有 $1 \le i \le n$，如果它按照除了第 $i$ 个医生之外的所有医生的药方吃药，它总共将花费多少钱。

## 【输入格式】

第一行两个正整数 $n, m$，分别为医生数和药片种类数。

接下来一行 $m$ 个正整数 $c_1 \sim c_m$。

接下来 $n$ 行，第 $i$ 行描述第 $i$ 个医生。首先三个正整数 $L_i, R_i, K_i$，后面 $K_i$ 个正整数 $x_{i,1}, x_{i,2}, \ldots, x_{i,K_i}$。保证 $x_{i,1}, x_{i,2}, \ldots, x_{i,K_i}$ 互不相同。

## 【输出格式】

输出 $n$ 个非负整数，第 $i$ 个表示，如果尼特认为第 $i$ 个医生是庸医并除开他的药方，它总共将花费多少钱。

## 【样例 1 输入】

```
5 4
10000 1000 100 10
3 4 2 2 3
4 8 3 1 2 4
6 7 2 3 4
8 9 2 1 4
2 6 3 1 2 3
```

## 【样例 1 输出】

```
87660 75640 87560 77650 66460
```

**【样例 1 解释】**

这里仅解释输出中的第一个和第五个数。

如果第一位医生是庸医，则尼特：

- 在第 $2,3$ 天会吃药片 $1,2,3$。花费 $11100 \times 2 = 22200$ 元。
- 在第 $4,5,6,7$ 天会吃药片 $1,2,3,4$。花费 $11110 \times 4 = 44440$ 元。
- 在第 $8$ 天会吃药片 $1,2,4$。花费 $11010$ 元。
- 在第 $9$ 天会吃药片 $1,4$。花费 $10010$ 元。

总花费 $87660$ 元。

如果第五位医生是庸医，则尼特：

- 在第 $3$ 天会吃药片 $2,3$。花费 $1100$ 元。
- 在第 $4$ 天会吃药片 $1,2,3,4$。花费 $11110$ 元。
- 在第 $5$ 天会吃药片 $1,2,4$。花费 $11010$ 元。
- 在第 $6,7$ 天会吃药片 $1,2,3,4$。花费 $11110 \times 2 = 22220$ 元。
- 在第 $8$ 天会吃药片 $1,2,4$。花费 $11010$ 元。
- 在第 $9$ 天会吃药片 $1,4$。花费 $10010$ 元。

总花费 $66460$ 元。

**【样例 2】**

见选手目录下 ***doctor/doctor2.in*** 与 ***doctor/doctor2.ans***。样例 2 满足子任务 1 的性质。

**【样例 3】**

见选手目录下 ***doctor/doctor3.in*** 与 ***doctor/doctor3.ans***。样例 3 满足子任务 3 的性质。

**【样例 4】**

见选手目录下 ***doctor/doctor4.in*** 与 ***doctor/doctor4.ans***。样例 4 满足子任务 5 的性质。

**【子任务】**

本题捆绑测试，你需要通过一个子任务的所有测试点才能得到子任务的分数。

对于所有数据：$1 \le n, m \le 5 \times 10^5, 1 \le L_i \le R_i \le 10^6, 1 \le K_i \le m, \sum K_i \le 10^6$。

| 子任务编号 | 特殊性质 | 分数 |
|---|---|---|
| 1 | $n \leq 100, m \leq 100, R_i \leq 100, \sum K_i \leq 100$ | 20 |
| 2 | $n \leq 5000, m \leq 5000, R_i \leq 5000, \sum K_i \leq 10^4$ | 20 |
| 3 | $[L_i, R_i]$ 互不相交 | 20 |
| 4 | $m = 1$ | 20 |
| 5 | 无 | 20 |

# 拓扑序计数（topo）

## 【题目描述】

本题中涉及到的图论定义：

- 一个 $n$ 个点，点的编号为 $1,2,\ldots,n$ 的有向图 $G=(V,E)$ 的**拓扑序**是一个 $1,2,\ldots,n$ 的排列 $p$，且若 $E$ 中存在 $x\to y$ 的边，就有 $p$ 中 $x$ 出现在 $y$ 之前。

今天，算法竞赛机器人小 G 学习了拓扑排序相关知识。凭着强大的机器学习本领，它很快便一并学会了如何计算一个有向无环图的拓扑序个数。接着，它开始思考一个拓展问题：给定一个有向无环图 $G$ 和两个 $G$ 中的点 $u,v$，请你求出有多少种 $G$ 的拓扑序满足 $u$ 排在 $v$ 之前。

你知道稍加思考后小 G 也能秒掉这题。不巧，就在这时停电了，依靠插头进食的小 G 也因此停止工作了。所以你只好自己解决这个拓展问题了。

为了让问题更富有挑战性，设 $G$ 中总点数为 $n$，请你对所有 $n(n-1)$ 对 $(u,v)$ 都求出答案。

## 【输入格式】

本题有多组数据，第一行是数据组数 $T$。

对于每组数据：第一行两个正整数 $n,m$，分别为 $G$ 的点数和边数。接下来 $m$ 行，每行两个正整数 $x,y$，表示有向图里一条 $x\to y$ 的边。保证没有重边且 $x<y$（也就是 $[1,2,\ldots,n]$ 总是一个合法拓扑序）。

**保证同一个测试点中至多有 5 组数据满足** $n>10$。

## 【输出格式】

对每组数据输出一个 $n\times n$ 的矩阵，第 $i$ 行第 $j$ 列是 $v=i,u=j$ 时的答案，注意 $(v,u)$ 的顺序和 $(i,j)$ 是反的。特别地，当 $i=j$ 时请你输出 0。

## 【样例 1 输入】

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

**【样例 1 输出】**

```
0 0 0
2 0 1
2 1 0
0 0 3 1
6 0 5 3
3 1 0 0
5 3 6 0
```

**【样例 1 解释】**

对于第一组数据，原图共有两种拓扑序 $[1,2,3],[1,3,2]$。满足 1 在 2 前面的有 2 种，所以答案矩阵的第 2 行第 1 列是 2；满足 3 在 2 前面的有 1 种，所以答案矩阵的第 2 行第 3 列是 1。

**【样例 2】**

见选手目录下 ***topo/topo2.in*** 与 ***topo/topo2.ans***。样例 2 满足子任务 1 的性质。

**【样例 3】**

见选手目录下 ***topo/topo3.in*** 与 ***topo/topo3.ans***。样例 3 满足子任务 10 的性质。

**【子任务】**

本题捆绑测试，你需要通过一个子任务的所有测试点才能得到子任务的分数。

对于所有数据：$1 \le T \le 100, 1 \le n \le 20, 0 \le m \le \binom{n}{2}$，**保证同一个测试点中至多有 5 组数据满足** $n > 10$。

| 子任务编号 | $n \leq$ | $m \leq$ | $T \leq$ | 分值 |
|---|---|---|---|---|
| 1 | 5 | $\binom{n}{2}$ | 20 | 10 |
| 2 | 20 | 0 | | 5 |
| 3 | | 1 | | 5 |
| 4 | | 2 | | 5 |
| 5 | | 10 | | 10 |
| 6 | 10 | $\binom{n}{2}$ | 30 | 5 |
| 7 | 12 | | 40 | 5 |
| 8 | 14 | | 50 | 10 |
| 9 | 16 | | 60 | 5 |
| 10 | 17 | | 70 | 5 |
| 11 | 18 | | 80 | 10 |
| 12 | 19 | | 90 | 5 |
| 13 | 20 | | 100 | 20 |

# 序列（loose）

## 【题目描述】

今天是 YQH 的生日，她得到了一个长度为 $n$ 的整数序列 $a_1, a_2, \ldots, a_n$ 作为生日礼物。

然而，YQH 并不对这个序列满意，因为这个序列可能并不合法。

一个序列 $\{a_i\}$ 合法，当且仅当 $\max\limits_{i=1}^{n}\{a_i\} + \min\limits_{i=1}^{n}\{a_i\} > n$，其中 $n$ 为序列长度，特别的，我们规定 $\varnothing$ 是合法的。

为了让 YQH 满意，你需要找到一个 $a_1, a_2, \ldots, a_n$ 的一个子段，使得这个子段是合法的。一个序列 $b_1, b_2, \ldots, b_m$ 是 $a_1, a_2, \ldots, a_n$ 的子段当且仅当 $b_1, b_2, \ldots, b_m$ 可以由 $a_1, a_2, \ldots, a_n$ 删掉若干个(可以为 0)开头及结尾的元素得到，比如 $[2,3], [1,2], [3,4], [1,2,3,4], \varnothing$ 都是 $[1,2,3,4]$ 的子段。

符合条件的子段可能很多，所以 YQH 只想要你找到，$a_1, a_2, \ldots, a_n$ 的所有合法子段的长度的最大值。

然而，YQH 得到的序列有魔力，所以它会产生变化，YQH 希望你对于初始的以及每次变化后的 $\{a_i\}$ 都求出答案。

## 【输入格式】

第一行一个正整数及一个非负整数 $n, m$。其中 $m$ 是 $\{a_i\}$ 的变化次数。

第二行 $n$ 个整数表示初始的 $a_1, a_2, \ldots, a_n$。

接下来，描述 $m$ 次变化，每次变化由若干行描述：

其中第一行一个非负整数表示 $k$，接下来 $k$ 行，第 $i$ 行两个正整数 $x_i, y_i$。假如变化前的序列为 $\{a_i\}$，那么对 $\{a_i\}$ 依次交换 $(a_{x_1}, a_{y_1}), (a_{x_2}, a_{y_2}), \ldots, (a_{x_k}, a_{y_k})$，得到的序列 $\{a'_i\}$ 就是变化后的序列。

变化之间**不独立**（见样例解释）

## 【输出格式】

第一行一个整数表示初始 $\{a_i\}$ 的答案。

接下来 $m$ 行，第 $i$ 行一个整数表示第 $i$ 次变化后的答案。

## 【样例 1 输入】

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

```
1
1 2
```

**【样例 1 输出】**

```
2
3
4
```

$\{a_i\}$ 初始为 $[1, 2, -2, 3, 4]$，其中一个合法子段为 $[3, 4]$。

第一次变化，交换 $(a_2, a_3)$ 后 $\{a_i\}$ 为 $[1, -2, 2, 3, 4]$，其中一个合法子段为 $[2, 3, 4]$。

第二次变化，交换 $(a_1, a_2)$，$\{a_i\}$ 为 $[-2, 1, 2, 3, 4]$，其中一个合法子段为 $[1, 2, 3, 4]$。

**【样例 2 输入】**

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

**【样例 2 输出】**

```
2
2
1
```

$\{a_i\}$ 初始为 $[-1, -1, 2, 1, 1]$，其中一个合法子段为 $[2, 1]$。

第一次变化，先交换 $(a_3, a_4)$，再交换 $(a_2, a_5)$，最终 $\{a_i\}$ 为 $[1, 1, 1, 2, -1]$，其中一个合法子段为 $[1, 2]$。

第二次变化，先交换 $(a_2, a_5)$，再交换 $(a_1, a_4)$，最终 $\{a_i\}$ 为 $[2, -1, 1, 1, 1]$，其中一个合法子段为 $[1]$。

**【样例 3 输入/输出】**

见选手目录下 ***loose/loose3.in*** 与 ***loose/loose3.ans***。

**【样例 4 输入/输出】**

见选手目录下 ***loose/loose4.in*** 与 ***loose/loose4.ans***。

**【数据范围】**

令所有变化的交换次数 $k$ 之和为 $K$。
对于所有数据，保证 $1 \le n \le 10^6, 0 \le m \le 30, 0 \le K \le 10^6, |a_i| \le 10^9, x_i \ne y_i$。

| 子任务编号 | $n \le$ | $m \le$ | 子任务分值 |
|---|---|---|---|
| 1 | 2000 | 30 | 20 |
| 2 | $2 \times 10^5$ | 2 | 20 |
| 3 | $10^6$ | 2 | 20 |
| 4 | $10^6$ | 30 | 40 |

# 水果（fruit）

## 【题目描述】

你在超市里工作。

超市里有 $n$ 个水果，第 $i$ 个水果的美味度为 $i$ ，价格为 $c_i$ ，并且保证 $c_i$ 单调不降。

现在要把这 $n$ 个水果摆成一排放到货架上，第 $j$ 个位置摆的水果是 $a_j$ 。但是你还没想好摆的顺序，所以可能会有 $a_j = -1$ ，表示这个位置摆的水果未定。

在你决定了摆放顺序之后，一位顾客进来买水果。他会从第一个位置开始往后走，每当遇到一个美味度比之前都要高的水果时就会把它买下，直到看完第 $k$ 个位置后离开。

你希望选择一个最优的摆放顺序，使得这位顾客出的钱最多。

但是你并不知道 $k$ 是多少，因此你希望对每个 $k$ 都求出答案。你对不同的 $k$ 给出的顺序可以不同。

## 【输入格式】

第一行一个正整数 $n$ 。

第二行 $n$ 个整数 $a_1, a_2, \cdots, a_n$ 。

第三行 $n$ 个正整数 $c_1, c_2, \cdots, c_n$ 。

## 【输出格式】

输出一行 $n$ 个正整数，表示 $k = 1, 2, \cdots, n$ 时的答案。

## 【样例 1 输入】

```
5
-1 3 -1 -1 -1
1 2 2 2 3
```

## 【样例 1 输出】

```
3 4 7 9 9
```

**【样例 1 解释】**

- $k = 1$ 的最优方案是把第 5 个水果放第一个位置，即令 $a_1 = 5$ ，后面任意。
- $k = 2$ 的最优方案是令 $a_1 = 2$ 。
- $k = 3$ 的最优方案是令 $a_1 = 2, a_3 = 5$ 。
- $k = 4$ 的最优方案是令 $a_1 = 2, a_3 = 4, a_4 = 5$ 。
- $k = 5$ 的最优方案是令 $a_1 = 2, a_3 = 4, a_4 = 5, a_5 = 1$ 。

**【样例 2 输入】**

```
13
-1 -1 5 6 -1 -1 7 11 -1 -1 10 -1 -1
1 1 1 1 1 1 1 1 1 1 1 1 1
```

**【样例 2 输出】**

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

**【样例 3 输入】**

```
10
-1 -1 -1 -1 5 -1 -1 -1 9 -1
5 11 24 27 35 60 72 81 91 92
```

**【样例 3 输出】**

```
92 173 245 305 305 332 356 367 406 498
```

**【样例 4】**

见选手目录下 ***fruit/fruit4.in*** 与 ***fruit/fruit4.ans***。样例 4 满足子任务 3 的性质。

**【样例 5】**

见选手目录下 ***fruit/fruit5.in*** 与 ***fruit/fruit5.ans***。样例 5 满足子任务 4 的性质。

**【样例 6】**

见选手目录下 ***fruit/fruit6.in*** 与 ***fruit/fruit6.ans***。样例 6 满足子任务 5 的性质。

**【子任务】**

本题采用捆绑测试，你需要通过一个子任务的所有测试点才能得到子任务的分数。

对于所有数据，保证 $1 \leq n \leq 4 \times 10^5, -1 \leq a_i \leq n, a_i \neq 0, 1 \leq c_i \leq 10^9$ ，$a$ 中不存在两个相同的正整数，$c$ 单调不降。

| 子任务编号 | 特殊性质 | 分值 |
|---|---|---|
| 1 | $n \leq 8$ | 10 |
| 2 | $a_1 = a_2 = \cdots = a_n = -1$ | |
| 3 | $n \leq 200$ | 20 |
| 4 | $n \leq 2000$ | |
| 5 | $c_1 = c_2 = \cdots = c_n = 1$ | |
| 6 | 无 | |