

OIFHA 联考

时间：2024 年 3 月 10 日 12:00 ~ 08:00

题目名称	桃红复含宿雨	柳绿更带朝烟	花落家童未扫	莺啼山客犹眠
题目类型	传统型	传统型	传统型	传统型
可执行文件名	mutation.exe	farmland.exe	petal.exe	travel.exe
输入文件名	mutation.in	farmland.in	petal.in	travel.in
输出文件名	mutation.out	farmland.out	petal.out	travel.out
每个测试点时限	2.0 秒	1.0 秒	1.0 秒	2.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
测试点数目	10	20	25	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	mutation.cpp	farmland.cpp	petal.cpp	travel.cpp
-----------	--------------	--------------	-----------	------------

编译选项

对于 C++ 语言	-O2 -std=c++14 -Wl,--stack=536870912
-----------	--------------------------------------

桃红复含宿雨 (mutation)

【题目描述】

好雨知时节，雨后的桃花更加柔嫩可爱、芳香扑鼻。这让王维不禁回忆起了昨天晚上一个奇怪的梦境：

梦境里的桃树受到了神秘力量的侵袭，变异成了一张 n 个点 m 条边的无向连通图。这个图中有一些点是坏的，坏的点满足：

- 删除这个点及其相邻的边后，剩下的图会变成一棵树。这里的树指无简单环的无向连通图。

王维想要考考作为他的头号粉丝的你，他请你求出这张无向连通图中所有坏的点。

【输入格式】

从文件 `mutation.in` 中读入数据。

多组数据。第一行一个正整数 T ，表示数据组数。

每组数据的第一行，两个正整数 n, m ，分别表示图的点数和边数。点按 $1 \sim n$ 编号。

接下来 m 行，每行两个正整数 u_i, v_i ，表示这张图中有一条从 u_i 到 v_i 的边。数据保证无重边和自环，且图连通。

【输出格式】

输出到文件 `mutation.out` 中。

对于每组数据，输出一行若干个正整数，按从小到大的顺序输出所有坏的点的编号，相邻两个整数之间用一个空格隔开。数据保证每组数据至少存在 1 个坏的点。

【样例 1 输入】

1	2
2	6 6
3	1 2
4	1 4
5	2 3
6	3 4
7	3 5
8	3 6
9	6 7
10	1 2
11	1 4

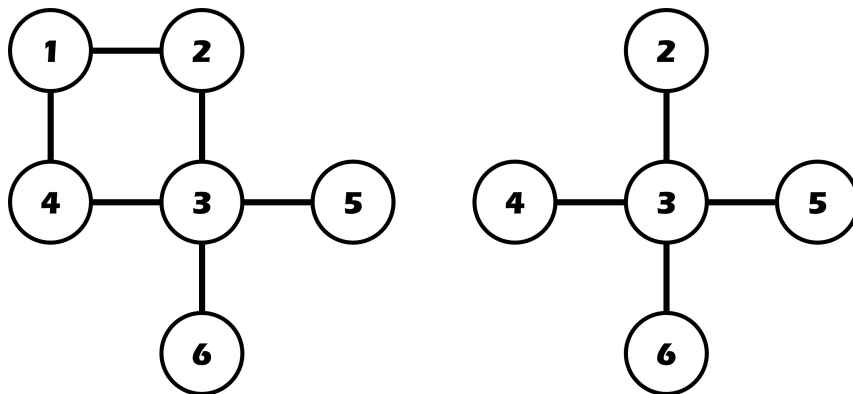
```
12 2 3
13 2 4
14 3 4
15 3 5
16 3 6
```

【样例 1 输出】

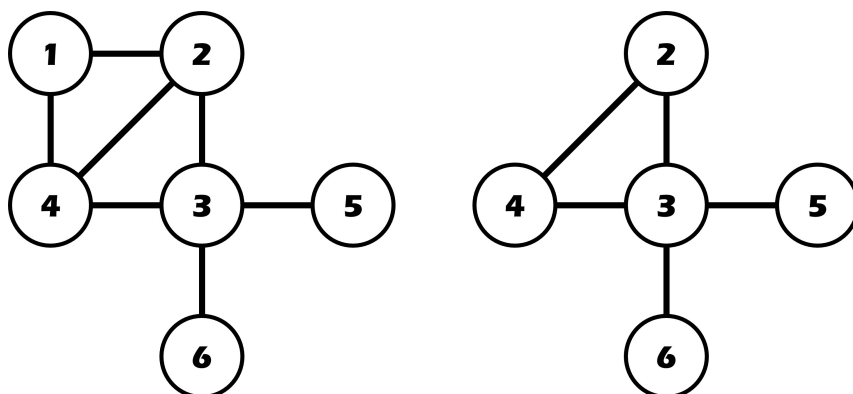
```
1 1 2 4
2 2 4
```

【样例 1 解释】

第一组数据对应的无向连通图如左下图，删除点 1 后如右下图。删除点 1 后的图是树，所以 1 是坏的点。



第二组数据对应的无向连通图如左下图，删除点 1 后如右下图。删除点 1 后的图不是树，所以 1 不是坏的点。



【样例 2】

见选手目录下的 *mutation/mutation2.in* 与 *mutation/mutation2.ans*。
该样例满足测试点 1, 2, 3 的限制。

【样例 3】

见选手目录下的 *mutation/mutation3.in* 与 *mutation/mutation3.ans*。
该样例满足测试点 4, 5 的限制。

【样例 4】

见选手目录下的 *mutation/mutation4.in* 与 *mutation/mutation4.ans*。
该样例满足测试点 6, 7 的限制。

【子任务】

测试点编号	$n, m \leq$	特殊性质
1,2,3	10^3	无
4,5	10^5	$m = n - 1$
6,7		$m = n$
8,9,10		无

对于所有数据, 满足 $1 \leq T \leq 20$, $1 \leq n, m \leq 10^5$ 。保证给出的图无重边和自环, 且图连通。保证每组数据至少存在 1 个坏的点。

柳绿更带朝烟 (farmland)

【题目描述】

缥缈的水气中，朦胧的柳枝碧绿而清幽。田野上忙碌充实的一天，便从此刻开始。

田野由 n 块农田组成，我们将这些农田按 $1 \sim n$ 编号。除了 1 号农田，其它每一块农田 i ($2 \leq i \leq n$) 都有通往 f_i 的单向道路。保证从任意一块农田出发，都可以沿着这些道路走到 1 号农田。

初始时， i ($2 \leq i \leq n$) 号农田里有 c_i 头牛，而王维想让所有牛都集中到 1 号农田。但是每条小路的承载量不同，从 i ($2 \leq i \leq n$) 号农田出发的小路的承载量是 m_i ，这表示 1 秒内最多允许 m_i 头牛经过这条小路。请注意，1 秒内同一头牛可以经过任意多条小路（也可以原地不动），农田没有容量限制。

q 次询问，每次询问王维会给你一个时间点 t ，请你告诉他 t 秒内最多有多少头牛能够到达 1 号农田？

【输入格式】

从文件 *farmland.in* 中读入数据。

第一行两个正整数 n, q ，分别表示农田的数量和询问的次数。

接下来 $n - 1$ 行，第 i 行三个正整数 $f_{i+1}, c_{i+1}, m_{i+1}$ ，含义如题目描述所示。

接下来 q 行，第 i 行一个正整数 t ，表示第 i 次询问王维给你的时间点。

【输出格式】

输出到文件 *farmland.out* 中。

输出共 q 行，每行一个整数，表示对应询问的答案。

【样例 1 输入】

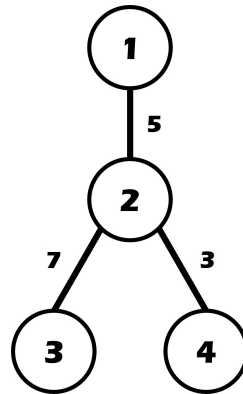
```
1 4 5
2 1 4 5
3 2 6 7
4 2 8 3
5 1
6 2
7 3
8 4
9 5
```

【样例 1 输出】

```
1 5
2 10
3 15
4 18
5 18
```

【样例 1 解释】

该样例描述的田野如下图所示：



一种最优策略如下：

- 初始时，每个农田中牛的数量分别为 0, 4, 6, 8。
- 第 1 秒，来自 2 号农田的 4 头牛，和来自 3 号农田的 1 头牛到达了 1 号农田。4 号农田还有 3 头牛到达了 2 号农田。每个农田中牛的数量分别为 5, 3, 5, 5。
- 第 2 秒，来自 3 号农田的 5 头牛到达了 1 号农田。4 号农田还有 3 头牛到达了 2 号农田。每个农田中牛的数量分别为 10, 6, 0, 2。
- 第 3 秒，来自 2 号农田的 5 头牛到达了 1 号农田。4 号农田还有 2 头牛到达了 2 号农田。每个农田中牛的数量分别为 15, 3, 0, 0。
- 第 4 秒，来自 2 号农田的 5 头牛到达了 1 号农田。每个农田中牛的数量分别为 18, 0, 0, 0。所有牛都在 4 秒内到达了 1 号农田，所以 4 秒后牛不再移动。

【样例 2 输入】

```
1 7 6
2 1 12 3
3 2 14 4
4 3 16 6
5 1 11 3
```

```

6 2 13 4
7 3 15 6
8 1
9 2
10 3
11 4
12 5
13 6

```

【样例 2 输出】

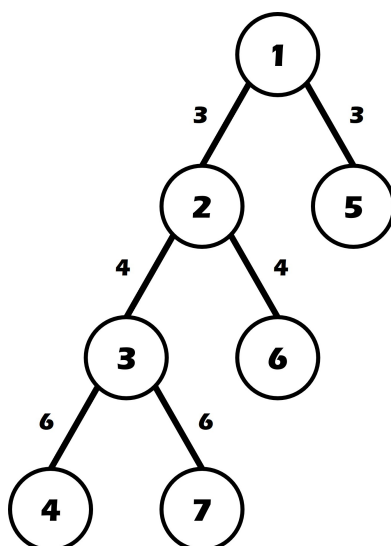
```

1 6
2 12
3 18
4 23
5 26
6 29

```

【样例 2 解释】

该样例描述的田野如下图所示：



【样例 3】

见选手目录下的 *farmland/farmland3.in* 与 *farmland/farmland3.ans*。
该样例满足测试点 8, 9, 10 的限制。

【样例 4】

见选手目录下的 *farmland/farmland4.in* 与 *farmland/farmland4.ans*。
该样例满足测试点 11, 12, 13, 14 的限制。

【样例 5】

见选手目录下的 *farmland/farmland5.in* 与 *farmland/farmland5.ans*。
该样例满足测试点 15, 16, 17, 18, 19, 20 的限制。

【子任务】

测试点编号	$n, q \leq$	特殊性质
1,2,3	200	所有 $t \leq 200$
4,5,6,7	2,000	无
8,9,10	2×10^5	$f_2 = 1, f_i = 2 \ (3 \leq i \leq n)$
11,12,13,14		$f_i = i - 1 \ (2 \leq i \leq n)$
15,16,17,18,19,20		无

对于所有数据，满足 $2 \leq n, q \leq 2 \times 10^5, f_i < i, 1 \leq c_i, m_i, t \leq 10^9$ 。

花落家童未扫 (petal)

【题目描述】

落红满地，似给这院落披上一件粉红的新衣；早起的王维见到这惹人怜惜的花瓣，心里有了一番打算。

王维将地上的花瓣捡起，排在一行 n 个位置上；从左到右的第 i 个位置上有 a_i 朵花瓣。王维想把这些位置分段，使得：

- 对于每一段 $l_j \sim r_j$ ，设 x 和 y 分别表示 $a_{l_j}, a_{l_j+1}, \dots, a_{r_j}$ 中的最小值和最大值，则需要满足 $x \leq r_j - l_j + 1 \leq y$ 。

王维想要考考作为他的头号粉丝的你，不同的合法分段方案的数量是多少？

形式化地讲，对于一个长度为 n 的序列 a_i ，我们定义一个分段方案为：

- 一个长为 k 的序列 r_j ，满足 $0 < r_1 < r_2 < \dots < r_k = n$ 。第 j 段即为 $r_{j-1} + 1 \sim r_j$ （定义 $r_0 = 0$ ）。记 $l_j = r_{j-1} + 1$ 。

我们称一个分段方案合法，当且仅当：对于所有 j ($1 \leq j \leq k$)，均有：

$$\min_{l_j \leq i \leq r_j} \{a_i\} \leq r_j - l_j + 1 \leq \max_{l_j \leq i \leq r_j} \{a_i\}$$

求不同的合法分段方案的数量。我们称两个分段方案不同，当且仅当它们的 k 不同，或 r_j 中的某一位不同。

【输入格式】

从文件 `petal.in` 中读入数据。

第一行一个正整数 n ，表示摆放花瓣的位置总数。

第二行 n 个正整数 a_i ，表示第 i 个位置上的花瓣数量。

【输出格式】

输出到文件 `petal.out` 中。

一行一个整数，表示不同的合法分段方案的数量。

【样例 1 输入】

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

【样例 1 输出】

```
1 5
```

【样例 1 解释】

合法的分段方案为:

- [2, 4], [1], [4, 3, 2]
- [2, 4], [1, 4], [3, 2]
- [2, 4], [1, 4, 3, 2]
- [2, 4, 1], [4, 3, 2]
- [2, 4, 1, 4], [3, 2]

【样例 2 输入】

```
1 8
2 1 3 1 1 1 3 1 1
```

【样例 2 输出】

```
1 19
```

【样例 3】

见选手目录下的 *petal/petal3.in* 与 *petal/petal3.ans*。
该样例满足测试点 3, 4, 5, 6 的限制。

【样例 4】

见选手目录下的 *petal/petal4.in* 与 *petal/petal4.ans*。
该样例满足测试点 7, 8, 9 的限制。

【样例 5】

见选手目录下的 *petal/petal5.in* 与 *petal/petal5.ans*。
该样例满足测试点 18, 19, 20 的限制。

【子任务】

测试点编号	$n \leq$	特殊性质
1,2	20	无
3,4,5,6	2,000	
7,8,9	10^5	所有 $a_i \leq 500$
10,11,12,13		所有 a_i 只存在 2 种不同数值
14,15,16,17		当 i 为偶数时, $a_i = n$
18,19,20		所有 a_i 在范围内均匀随机
21,22		无
23,24,25	5×10^5	

对于所有数据, 满足 $1 \leq a_i \leq n \leq 5 \times 10^5$ 。

莺啼山客犹眠 (travel)

【题目描述】

黄莺歌声婉转，春色袅娜迷人。王维乘兴而行，打算遍游一番他所在的山谷。

这片山谷里有 n 座村庄，从西到东依次编号为 $1 \sim n$ 。第 i 座村庄会说编号为 h_i 的方言；只有王维提前学会了编号为 h_i 的方言，他才可以经过这座村庄。因为两两村庄说的方言互不相同，所以保证 h_i 是一个 $1 \sim n$ 的排列。

同时，山谷里还有 m 位老者，第 i 位老者居住在第 p_i 座村庄，会说编号在 $a_i \sim b_i$ 中的所有方言（包括 a_i 和 b_i ，保证 $a_i \leq h_{p_i} \leq b_i$ ）。如果王维向他支付 c_i 个金币，他将教会王维这些方言。

王维想要考考作为他的头号粉丝的你，对于每个整数 i ($1 \leq i \leq m$)，请你求出，如果王维化身为第 i 位老者（即初始时他不花费金币学会了 $a_i \sim b_i$ 的方言，而其它方言还未学会），他是否能够通过学习方言，使得自己可以经过所有的 n 座村庄？如果能，他还想请你求出学习过程中花费的最小金币总数。

【输入格式】

从文件 `travel.in` 中读入数据。

第一行两个正整数 n, m ，分别表示村庄总数和老者总数。

第二行 n 个正整数 h_i ，表示第 i 座村庄说的方言。

接下来 m 行，每行四个正整数 p_i, a_i, b_i, c_i ，表示第 i 位老者的信息。

【输出格式】

输出到文件 `travel.out` 中。

输出共 m 行，第 i 行一个整数，表示如果王维化身为第 i 位老者，他在学习过程中花费的最小金币总数。若无论如何也无法经过所有的 n 座村庄，输出 `-1`。

【样例 1 输入】

```
1 3 3
2 3 1 2
3 2 1 1 20
4 2 1 2 50
5 3 2 3 80
```

【样例 1 输出】

```
1 130
2 80
3 -1
```

【样例 1 解释】

化身为第 1 位老者出发的过程如下：

- 初始时，王维在第 2 座村庄，学会的方言为 1。
- 在第 2 位老者处花费 50 个金币学习，王维学会的方言变成了 1,2。
- 到第 3 座村庄，在第 3 位老者处花费 80 个金币学习，王维学会的方言变成了 1,2,3。
- 王维可以遍游所有村庄了，总共花费了 $50 + 80 = 130$ 个金币。

化身为第 2 位老者出发的过程如下：

- 初始时，王维在第 2 座村庄，学会的方言为 1,2。
- 到第 3 座村庄，在第 3 位老者处花费 80 个金币学习，王维学会的方言变成了 1,2,3。
- 王维可以遍游所有村庄了，总共花费了 80 个金币。

化身为第 3 位老者出发时，因为无论如何也无法到达第 2 座村庄，所以王维不能遍游所有村庄。

【样例 2 输入】

```
1 7 5
2 7 5 2 4 6 3 1
3 4 4 7 80
4 2 1 5 40
5 6 3 6 50
6 7 1 3 20
7 1 3 7 60
```

【样例 2 输出】

```
1 -1
2 80
3 -1
```

4	130
5	40

【样例 3】

见选手目录下的 `travel/travel3.in` 与 `travel/travel3.ans`。
该样例满足测试点 5, 6, 7 的限制。

【样例 4】

见选手目录下的 `travel/travel4.in` 与 `travel/travel4.ans`。
该样例满足测试点 11, 12, 13, 14 的限制。

【样例 5】

见选手目录下的 `travel/travel5.in` 与 `travel/travel5.ans`。
该样例满足测试点 15, 16, 17, 18, 19, 20 的限制。

【子任务】

测试点编号	$n, m \leq$	特殊性质
1,2	8	无
3,4	60	$h_i = i$
5,6,7		无
8,9,10	300	$h_i = i$
11,12,13		无
14,15,16,17,18,19,20	2,000	

对于所有数据，满足 $1 \leq n, m \leq 2,000$ ， $1 \leq h_i, p_i, a_i, b_i \leq n$ ， $1 \leq c_i \leq 10^6$ 。保证 h_i 是一个 $1 \sim n$ 的排列，且 $a_i \leq h_{p_i} \leq b_i$ 。