2023 年重庆市第三届信息学友谊赛

The Third CQOI Friendship Competition

第四试 by CQYC

时间：2023 年 8 月 29 日 07:50～ 12:20

题目名称	月华	光与影	夏虫	万分之一的光
题目类型	传统型	传统型	传统型	传统型
目录	tsuki	dream	summer	light
可执行文件名	tsuki	dream	summer	light
输入文件名	tsuki.in	dream.in	summer.in	light.in
输出文件名	tsuki.out	dream.out	summer.out	light.out
每个测试点时限	2.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MB	512 MB	512 MB	512MB
测试点数目	25	25	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	tsuki.cpp	dream.cpp	summer.cpp	light.cpp

编译选项

对于 C++ 语言	-O2 -std=c++14 -static

注意事项（请仔细阅读）

1. 选手提交的源程序请**直接放在个人目录下**，无需建立子文件夹；
2. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
3. C++ 中函数 main() 的返回值类型必须是 int，值必须为 0。
4. **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**。
5. 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**。。
6. 程序可使用的栈空间大小与该题内存空间限制一致。
7. 在终端中执行命令 `ulimit -s unlimited` 可将当前终端下的栈空间限制放大，但你使用的栈空间大小不应超过题目限制。

8. 若无特殊说明，每道题的**代码大小限制为 100KB**。
9. 若无特殊说明，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。
10. 输入文件中可能存在行末空格，请选手使用更完善的读入方式（例如 scanf 函数）避免出错。
11. 直接复制 PDF 题面中的跨页样例，数据将带有页眉页脚，建议选手直接使用对应目录下的样例文件进行测试。
12. 使用 std::deque 等 STL 容器时，请注意其内存空间消耗。
13. 请务必使用题面中规定的的编译参数，保证你的程序在本机能够通过编译。此外**不允许在程序中手动开启其他编译选项**，一经发现，本题成绩以 0 分处理。
14. 题目难度不一定升序排序。
15. 很好很好的题，爱来自「依」。

月华（tsuki）

【题目描述】

一次不经意地抬头，「依」望着如水的月华入了神。她伸出手朝着夜空随意画了一个矩形，想知道这一小片夜空的色彩度为多少。

夜空可以看作大小为 $n \times m$ 的矩阵，从上到下分别为第 1 行到第 n 行，从左到右分别为第 1 列到第 m 列。

众所周知月华是由月光衍射而成的，所以每个位置的色彩度可能会不同。具体地，可以用 $a_{i,j}$ 表示第 i 行第 j 列这个位置的色彩度。

一个矩形的色彩度定义为这个矩形中每一个位置色彩度的积。

「依」会询问你 Q 次。由于答案可能非常大，你只需要输出 $\sum_{i=1}^{Q} i \oplus (ans_i \% P)$ 对 147744151 取模后的结果即可。其中 ans_i 表示第 i 次询问的答案，$\oplus$ 表示按位异或。

【输入格式】

从文件 ***tsuki.in*** 中读入数据。

第一行包含三个正整数 n, m, P，分别表示矩阵的大小和模数。

接下来一个 n 行 m 列的矩阵表示夜空的色彩度。

第 $n+2$ 行一个正整数 Q 表示询问次数。

接下来的 Q 行，第 i 行包含四个正整数 x_1, y_1, x_2, y_2 表示第 i 次「依」询问的区域是以 (x_1, y_1) 为左上角，以 (x_2, y_2) 为右下角的子矩阵。

【输出格式】

输出到文件 ***tsuki.out*** 中。

一行一个正整数表示 $(\sum_{i=1}^{Q} i \oplus (ans_i \% P)) \% 147744151$。

由于本题数据量较大，请使用较快的读入输出方式。

【样例 1 输入】

```
5 5 14
1 11 5 11 1
3 9 11 2 2
13 1 9 7 13
5 3 5 3 11
5 9 9 13 2
```

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

【样例 1 输出】

```
23
```

【样例 1 解释】

每一次答案分别为 $0, 13, 2, 0, 7$。
$(0 \oplus 1) + (13 \oplus 2) + (2 \oplus 3) + (0 \oplus 4) + (7 \oplus 5) = 23$。

【样例 2】

见选手目录下的 ***tsuki/ex_tsuki2.in*** 与 ***tsuki/ex_tsuki2.out***。
该样例数据满足测试点 $1 \sim 5$ 的限制。

【样例 3】

见选手目录下的 ***tsuki/ex_tsuki3.in*** 与 ***tsuki/ex_tsuki3.out***。
该样例数据满足测试点 $18 \sim 19$ 的限制。

【数据范围】

对于所有的数据，保证：$1 \leq n, m \leq 1000$，$1 \leq Q \leq 2 \times 10^6$，$1 \leq x_1, x_2 \leq n$，$1 \leq y_1, y_2 \leq m$，$1 \leq a_{i,j} < P \leq 10^9$。

测试点编号	$n, m \leq$	$Q \leq$	P
1∼ 5	200	200	$\leq 10^9$
6∼ 7	≤ 1000	10^5	≤ 20
8∼ 9		2×10^6	
10∼ 11		10^5	≤ 100
12∼ 13		2×10^6	
14∼ 15		10^5	$\leq 10^6$
16∼ 17		2×10^6	
18∼ 19		10^5	$= 998244353$
20∼ 21		2×10^6	
22∼ 23		10^5	$\leq 10^9$
24∼ 25		2×10^6	

光与影（dream）

【题目描述】

光在不同的场景中，发生着丰富的变化，由聚到散，由明到暗，由近及远..... 与此同时，影所呈现出的意象被不断重构，表现在造型、色彩、明暗度、空间感、环境氛围等不同方面。这场变幻莫测的视觉呈现，赋予空间极强的层次感。

「依」突然想起很久以前种下过一棵名为「晦」的树，它本质是一棵斯普雷树。为了避免你不了解斯普雷树，「依」向你介绍了起来。

斯普雷树（`splay tree`）是一种基于均摊的二叉树，支持多种基本操作。

Splay 是斯普雷树的一个基本操作。对树上结点 x 执行 $Splay(x)$ 时，将对 x 不断地进行斯普雷操作，直至将 x 旋转到根。

斯普雷操作有 3 种，对于被旋转结点 x 的不同情况，操作的类型也不同。描述如下：

1. Zig：当 x 的父结点 p 为根时，将 x 旋转一次，如下图：

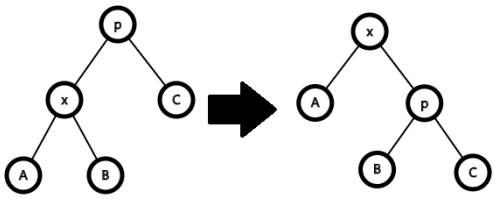

图 1: Zig

2. Zig-zig：当 x 的父结点 p 不为根，且 x 和 p 同时为各自父结点的左子结点或右子结点时，先旋转 p 一次，再旋转 x 一次，如下图：

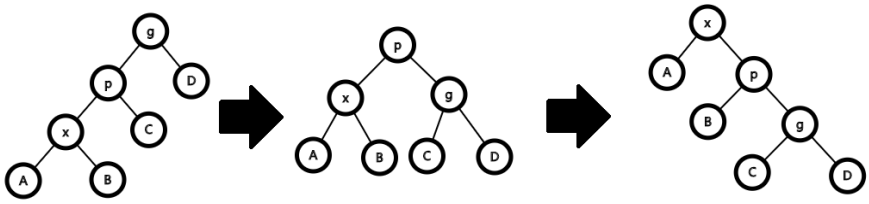

图 2: Zig-zig

3. Zig-zag：当 x 的父结点 p 不为根，且 x 和 p 一个是左子结点，一个是右子结点时，连续旋转 x 两次，如下图：

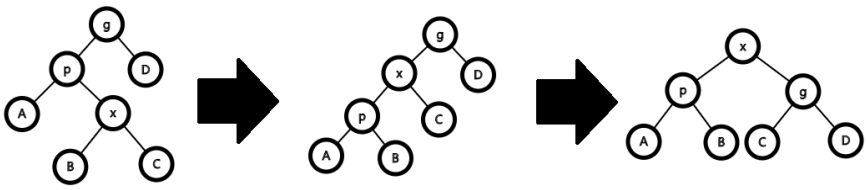

图 3: Zig-zag

设 ls 表示 x 左儿子，rs 表示 x 右儿子，那么旋转 x 可以简单描述为：

- 若 x 是 fa_x 的左儿子，那么令 rs 成为 fa_x 的左儿子，fa_x 成为 x 的右儿子。
- 若 x 是 fa_x 的右儿子，那么令 ls 成为 fa_x 的右儿子，fa_x 成为 x 的左儿子。

定义一个结点 x 的明亮度为 x 的深度。（根结点深度为 0 ）。

现在「依」要等概率随机一个结点 y 进行 $Splay(y)$ 。你能告诉她操作完后点 x 的期望明亮度为多少吗？

【输入格式】

从文件 ***dream.in*** 中读入数据。

第一行一个正整数 n 表示「晦」的结点个数。

接下来 n 行，每行两个非负整数，第 i 行表示结点 i 的左子结点和右子结点的编号，若为 0 则表示没有。其中 1 号点一定是根。

【输出格式】

输出到文件 ***dream.out*** 中。

输出共 n 行，每行一个非负整数。第 i 行表示操作完后结点 i 的期望明亮度乘上 n 对 10^9+7 取模的值。

由于本题数据量较大，请使用较快的读入输出方式。

【样例 1 输入】

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

【样例 1 输出】

```
5
5
8
10
8
```

【样例 1 解释】

下图是原树依次进行 $Splay(1), Splay(2)...Splay(5)$ 后的形态。

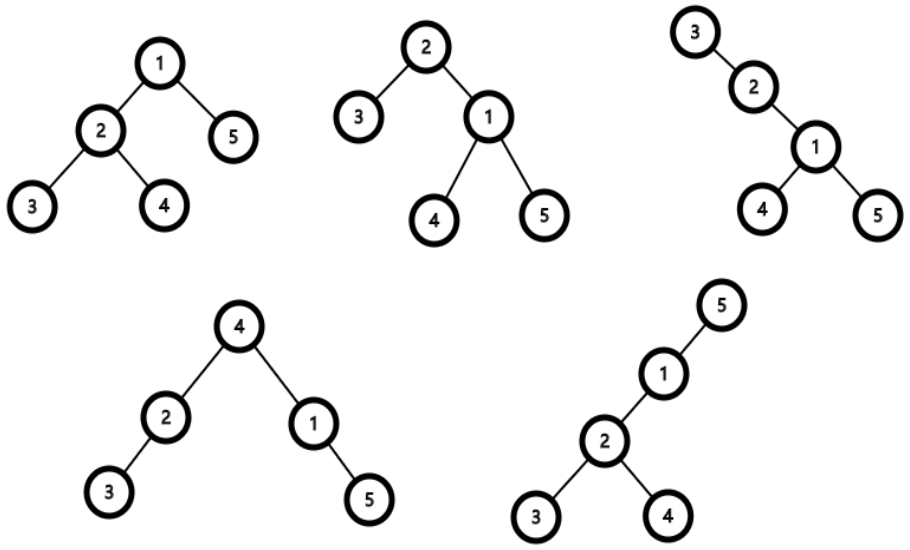

图 4: 样例 1 解释

1 的期望深度为 $\frac{0+1+2+1+1}{5}$；
2 的期望深度为 $\frac{1+0+1+1+2}{5}$；
3 的期望深度为 $\frac{2+1+0+2+3}{5}$；
4 的期望深度为 $\frac{2+2+3+0+3}{5}$；
5 的期望深度为 $\frac{1+2+3+2+0}{5}$。

【样例 2】

见选手目录下的 ***dream/ex_dream2.in*** 与 ***dream/ex_dream2.out***。
该样例数据满足测试点 $1 \sim 8$ 的限制。

【样例 3】

见选手目录下的 ***dream/ex_dream3.in*** 与 ***dream/ex_dream3.out***。
该样例数据满足测试点 $9 \sim 10$ 的限制。

【样例 4】

见选手目录下的 ***dream/ex_dream4.in*** 与 ***dream/ex_dream4.out***。
该样例数据满足测试点 $11 \sim 13$ 的限制。

【样例 5】

见选手目录下的 ***dream/ex_dream5.in*** 与 ***dream/ex_dream5.out***。
该样例数据满足测试点 $20 \sim 22$ 的限制。

【测试点约束】

对于所有数据，保证：$1 \leq n \leq 10^6$。

测试点编号	$n \leq$	特殊限制
$1 \sim 8$	2000	无
$9 \sim 10$	2×10^5	A
$11 \sim 13$	10^5	B
$14 \sim 16$		C
$17 \sim 19$		无
$20 \sim 22$	5×10^5	
$23 \sim 25$	10^6	

- 特殊限制 A：二叉树为满二叉树。
- 特殊限制 B：所有节点均没有右子节点。
- 特殊限制 C：所有节点均只有至多一个子节点。

夏虫（summer）

【题目描述】

蝴蝶经历化蛹之痛诞生翅膀，蝉在地下蛰伏以求破土光明。

夏虫们只为在仅此一次的盛夏去追逐最炽热的骄阳。我们结伴而行，纵情歌唱，一同触碰未知的绚烂奇迹！

在每次夏日里，「依」都会和「绫」在公路旁捕夏虫。具体地，公路可看作一条直线，有 n 只夏虫藏在路边。夏虫从左往右，用 1 到 n 依次编号。其中第 i 只夏虫的狡猾度为 a_i 。

「依」和「绫」每次总要把 n 只夏虫全部捉住才肯善罢甘休。最开始时，她们会走到某只夏虫 x 面前，并将捕虫网的初始能力值设为 a_x 。

假设当前已经捕捉了所有编号在 $[l,r]$ 之间的夏虫，捕虫网能力值为 v，可以进行如下操作：

- 若 $l \neq 1$ 并且 $a_{l-1} \leq v$ ，那么她们可以花费 1 单位精力捕捉 $l-1$ 号夏虫；
- 若 $r \neq n$ 并且 $a_{r+1} \leq v$ ，那么她们可以花费 1 单位精力捕捉 $r+1$ 号夏虫；
- 花费 k 单位精力使得捕虫网的能力值为 $\min\{a_{l-1}, a_{r+1}\}$。在这里定义 $a_0 = a_{n+1} = 2012^{7^{12}}$。

每天都会有两只相邻的夏虫交换位置，「依」和「绫」每天也都会走到不同的夏虫面前。但是她们还需要省下精力来干别的事儿呢，所以求求你帮她们求出最少需要花费多少精力才能捕捉到所有夏虫吧！

【输入格式】

从文件 ***summer.in*** 中读入数据。

第一行两个整数 n, k，分别表示夏虫个数和更改捕虫网能力值需要花费的精力。

第二行 n 个整数，第 i 个数 a_i 表示编号为 i 的夏虫的狡猾度。

第三行一个整数 Q ，表示夏日会持续 Q 天。

接下来 Q 行，每行三个整数 x_i, l_i, r_i，表示在第 i 天，编号为 x_i 的夏虫将会和编号为 x_i+1 的夏虫交换位置。（**交换后将按照从左往右的顺序重新为夏虫编号**）。并询问若「依」和「绫」等概率出现在编号在 $[l_i, r_i]$ 之间的夏虫的面前，抓完所有夏虫期望花费的最少精力与 $(r_i - l_i + 1)$ 相乘的结果。不难证明这是一个整数。

【输出格式】

输出到文件 ***summer.out*** 中。

输出 Q 行，第 i 行一个整数表示第 i 天期望花费最少精力与 $(r_i - l_i + 1)$ 相乘的结果。

【样例 1 输入】

```
7 7
4 3 2 1 2 3 4
5
1 1 3
2 3 5
3 2 5
4 1 5
5 1 5
```

【样例 1 输出】

```
39
53
73
86
93
```

【样例 1 解释】

以第一天为例，交换 (a_1, a_2) 后序列变为 $\{3, 4, 2, 1, 2, 3, 4\}$。
若出现在位置 1，需要耗费的精力为 $6 + 7 = 13$；
若出现在位置 2，需要耗费的精力为 6；
若出现在位置 3，需要耗费的精力为 $6 + 14 = 20$；
期望消耗精力为 $\frac{39}{3} = 13$，乘上区间长度可得答案为 $13 \times 3 = 39$。

【样例 2】

见选手目录下的 ***summer/ex_summer2.in*** 与 ***summer/ex_summer2.out***。
该样例数据满足测试点 4 的限制。

【样例 3】

见选手目录下的 ***summer/ex_summer3.in*** 与 ***summer/ex_summer3.out***。

该样例数据满足测试点 5 ∼ 6 的限制。

【样例 4】

见选手目录下的 ***summer/ex_summer4.in*** 与 ***summer/ex_summer4.out***。
该样例数据满足测试点 9 ∼ 10 的限制。

【样例 5】

见选手目录下的 ***summer/ex_summer5.in*** 与 ***summer/ex_summer5.out***。
该样例数据满足测试点 16 ∼ 18 的限制。

【测试点约束】

对于所有数据，保证：$1 \le n \le 10^5$，$1 \le k \le 10^6$，$1 \le a_i \le 10^9$，$1 \le Q \le 10^5$，$1 \le x_i < n$，$1 \le l_i \le r_i \le n$。

测试点编号	$n \le$	$Q \le$	特殊性质
1	100	100	D
2			无
3	1000	1000	D
4			无
5 ∼ 6	10^5	10^5	AD
7 ∼ 8			A
9 ∼ 10			BD
11 ∼ 12			B
13 ∼ 15			C
16 ∼ 18			D
19 ∼ 20			无

- 特殊性质 A：保证 $x_1 = x_2 = ... = x_Q$。
- 特殊性质 B：保证 $l_i = r_i$。
- 特殊性质 C：保证 $a_i \le 10$。
- 特殊性质 D：保证 a_i 互不相同。

万分之一的光（light）

【题目描述】

“你知道吗，听说星星是死去的萤火虫，萤火虫是还活着的星星哦。”

「依」和「绫」漫步在夏夜的草丛间，试图捕获这纷飞的夏日萤火。

草丛里共有 n 只萤火虫，有 $n-1$ 对萤火虫可以相互通信。且保证任意两只萤火虫都能直接或间接通信。

若「依」和「绫」决定捕捉编号为 i 的萤火虫，那么她们就能捕获 a_i 单位的光。且编号为 i 的萤火虫将会传递信息给能和它直接通信的，尚未被捕捉的萤火虫。

当编号为 j 的萤火虫收到来自编号为 i 的萤火虫的信息后，会令 $a_j = a_j + a_i$ 。

现在「依」和「绫」想知道在所有 $n!$ 种捕捉萤火虫的顺序中，最多能捕获多少单位的光。

【输入格式】

从文件 ***light.in*** 中读入数据。

第一行一个正整数 n 表示萤火虫个数。

第二行 n 个整数，第 i 个整数表示 a_i 。

接下来 $n-1$ 行，每行两个整数 x_i, y_i 表示编号为 x_i 的萤火虫与编号为 y_i 的萤火虫能直接通信。

【输出格式】

输出到文件 ***light.out*** 中。

一行一个整数表示答案。

【样例 1 输入】

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

【样例 1 输出】

```
47
```

【样例 1 解释】

一种可能的捕捉顺序为 $\{4,5,2,1,3\}$，所捕获的光为 $4+5+11+12+15=47$ 单位。

【样例 2】

见选手目录下的 ***light/ex_light2.in*** 与 ***light/ex_light2.out***。
该样例数据满足测试点 $1\sim2$ 的限制。

【样例 3】

见选手目录下的 ***light/ex_light3.in*** 与 ***light/ex_light3.out***。
该样例数据满足测试点 $3\sim4$ 的限制。

【样例 4】

见选手目录下的 ***light/ex_light4.in*** 与 ***light/ex_light4.out***。
该样例数据满足测试点 $5\sim7$ 的限制。

【样例 5】

见选手目录下的 ***light/ex_light5.in*** 与 ***light/ex_light5.out***。
该样例数据满足测试点 $8\sim10$ 的限制。

【样例 6】

见选手目录下的 ***light/ex_light6.in*** 与 ***light/ex_light6.out***。
该样例数据满足测试点 $14\sim16$ 的限制。

【测试点约束】

对于所有数据，保证：$1\le n\le 400$，$|a_i|\le 10^9$。

测试点编号	$n \leq$	特殊性质
$1 \sim 2$	10	无
$3 \sim 4$	19	
$5 \sim 7$	≤ 50	A
$8 \sim 10$		无
$11 \sim 13$	≤ 400	A
$14 \sim 16$		B
$17 \sim 20$		无

- 特殊性质 A：保证 $\forall 1 \leq i \leq n-1, x_i = i+1, y_i = i$。
- 特殊性质 B：保证 $\forall 1 \leq i \leq n-1, x_i = 1, y_i = i+1$。