

第 1 屆 緹 觀 鄴 禱 啤 俁 冴 婆 鄴 菱 計

CCC-WF

第一試

cdqz

时间：2021 年 1 月 12 日

一、题目概况

题目名称	枇杷树	上古遗迹	吞天得手
题目类型	传统型	传统型	传统型
目录	loquat	relics	tt ds
源程序文件名	loquat.cpp	relics.cpp	tt ds.cpp
可执行文件名	loquat	relics	tt ds
输入文件名	loquat.in	relics.in	tt ds.in
输出文件名	loquat.out	relics.out	tt ds.out
每个测试点时限	1s	1s	1s
内存限制	512MB	512MB	512MB
子任务数目	4	4	4
编译选项	-lm -O2		

二、注意事项

1. 选手提交的源文件应直接存放在对应的个人文件夹中，无需建立对应题目的子文件夹。
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. 函数 `main()` 的返回值类型必须是 `int`，值必须为 `0`。
4. 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。
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。
8. 请注意题目按照目录名字典序排序，不一定按照题目难度排序。

枇杷树 (loquat)

【题目背景】

杨吞天家有个很大的后院，但是光秃秃的，看起来很是荒凉。杨吞天决定种一个枇杷树种子。许多年后，杨吞天的后院已经长满了“亭亭如盖”的枇杷树。

【题目描述】

杨吞天观察了枇杷树林的生长规律，他惊奇地发现，每一棵长出的枇杷树和之前长出的枇杷树有着令人惊奇的相似性！

具体来说，每一年都会新长出一颗枇杷树，第 i 年长出的枇杷树记为 T_i 。第 0 年长出来的枇杷树 T_0 由于只是一个种子，因此它是一棵只有一个结点的树，该结点编号为 0。

对于每个 T_i ($i > 0$)，都可以用 5 个参数来描述它， x_i, y_i, u_i, v_i, w_i ($x_i, y_i < i$)。表示将 T_{x_i} 的 u_i 号结点和 T_{y_i} 的 v_i 号结点用一条长度为 w_i 的边连结起来得到了 T_i (注意， x_i 可以等于 y_i ，并且生成新树的时候用了两棵树)。

杨吞天将每年的树上的结点进行了重新编号，具体来说：对于第 i 年长出来的枇杷树，在第 x_i 棵树中点的编号不会变，在 y_i 棵树中的点的编号加上第 x_i 棵树的大小作为新的编号。

杨吞天很喜欢吃枇杷，每棵树上的每个结点都有一个枇杷。他想知道每棵树上任意两个枇杷间的距离之和。形式化地，对于每一棵树，你要求

$$\sum_{i=0}^{n-1} \sum_{j=i+1}^{n-1} \text{dis}(i, j).$$

其中 n 为树的结点个数， $\text{dis}(i, j)$ 为结点 i, j 两个结点的简单路径上的边的长度和，注意结点从 0 开始编号。

由于答案可能很大，你只需要输出答案对 $10^9 + 7$ 取模的结果就可以了。

【输入格式】

从文件 `loquat.in` 中读入数据。

输入共若干行。

第一行一个正整数 m ，表示有 m 年。

接下来 m 行，每行 5 个正整数 x_i, y_i, u_i, v_i, w_i 描述 T_i 。

【输出格式】

输出到文件 `loquat.out` 中。

输出一共 m 行，每行一个非负整数，第 i 行的输出表示 T_i 对应的答案对 $10^9 + 7$ 取模的结果。

【样例 1 输入】

```
1 4
2 0 0 0 0 3
3 1 0 1 0 4
4 2 1 0 1 5
5 1 1 0 0 2
```

【样例 1 输出】

```
1 3
2 14
3 76
4 26
```

【样例 2】

见选手目录下的 *loquat/loquat2.in* 与 *loquat/loquat2.ans*。

【测试点约束】

对于所有测试点，满足 $1 \leq m \leq 300$ ， $0 \leq x_i, y_i < i$ ， $1 \leq w_i \leq 10^9$ ， $0 \leq u_i, v_i \leq 10^{18}$ ，保证树的大小不超过 2×10^{18} 。

每个子任务的具体限制见下表：

子任务编号	分值	特殊限制
1	10	$m \leq 10$
2	20	每棵树的大小不超过 10^5
3	20	树的形态总是链
4	50	无

上古遗迹 (relics)

【题目背景】

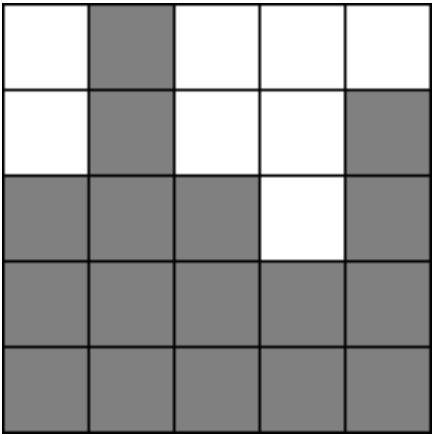
“沙……沙……沙……” 独行者的脚步一次次被刻进沙漠中，干冷的风携沙尘在男子的四周穿过。

“该死……这沙尘什么时候才能消停会儿……” 男子止不住地咳嗽，随即停了下来，开始查看便携式投影设备上的信息，“应该就在附近了才对……”

就在这时，风尘似乎是要止住，又仿佛是无意地，将不远处那方遗迹显现出来。

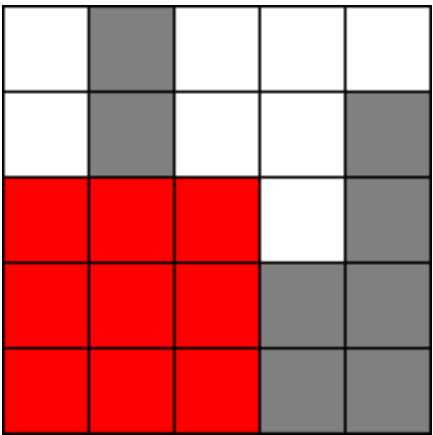
【题目描述】

ConneR 所寻找的遗迹是一排高耸的石板。这排石板总共由 n 块宽度均为 1 并且无缝隙地竖直矗立在地面上的石板组成，其中从左到右数的第 i 块石板高度为 h_i 。例如下图是 $n = 5$ 时这些石板可能的样子（图中每个小正方形面积为 1）：



ConneR 想要在石板上采集到尽可能多的信息，但他发现自己的信息采集装置的电量只够他采集一块矩形区域的石板信息了。同时，由于信息采集装置的设计缺陷，该装置采集的矩形区域必须是完整的，

ConneR 可以选择采集左下角 3×3 的一块矩形的信息（下图的红色部分），但不能选择采集左下角一块 3×4 的矩形的信息，因为第 4 块石板高度只有 2。



ConneR 想要问你他在这个遗迹上最大可以采集到多大面积的信息。同时,由于风沙影响了 ConneR 的行动范围,ConneR 会给出你多组询问,每次询问他在一个区间的石板上最多能采集到多大面积的信息。

【输入格式】

从文件 *relics.in* 中读入数据。

输入若干行。

第一行两个正整数 n 和 m , 代表石板的个数和询问数。

第二行 n 个正整数, 第 i 个整数代表 h_i , 即第 i 块石板的高度。

接下来 m 行, 每行两个正整数 l 和 r , 代表询问在 l 到 r 这个区间内的石板上 (包含左右端点) 最多能采集到多大面积的信息, 保证 $l \leq r$ 。

【输出格式】

输出到文件 *relics.out* 中。

输出 m 行, 第 i 行一个整数, 代表第 i 个询问的答案。

【样例 1 输入】

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

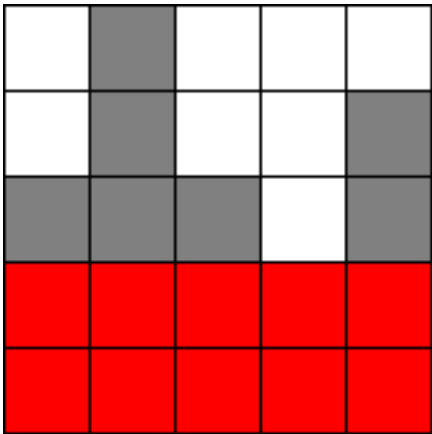
【样例 1 输出】

```
1 9
2 10
```

【样例 1 解释】

此样例和【题目描述】中给出的例子相同。

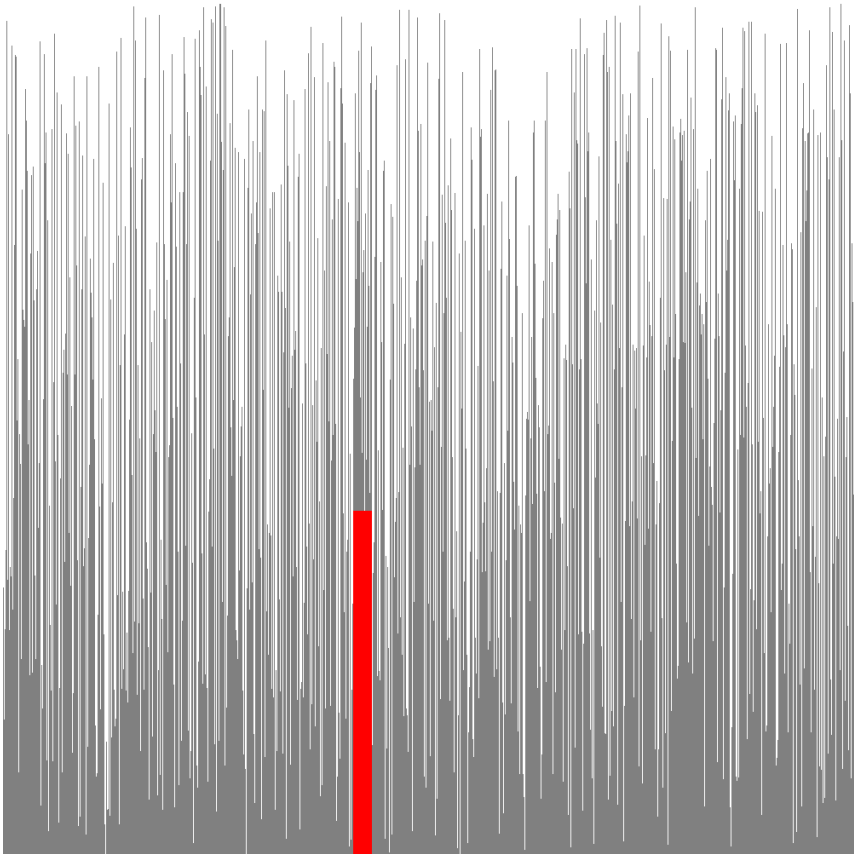
第一个询问选择的矩形即为【题目描述】中选择的矩形, 即左下角 3×3 的矩形; 第二个询问选择的矩形如下图红色矩形所示。



【样例 2】

见选手目录下的 *relics/relics2.in* 与 *relics/relics2.ans*。

【样例 2 解释】



见选手目录下的 *relics/relics2.png*

这些石板如上图所示。可以在从左到右数第 412 块石板至第 433 块石板上高为 404 米的矩形处采集到整排石板中最大面积 8888 的信息，即图中的红色矩形。所以对于所有 $l \leq 412$ 且 $r \geq 433$ 的询问，其答案都为 8888。

【样例 3】

见选手目录下的 *relics/relics3.in* 与 *relics/relics3.ans*。

【样例 3 解释】

此样例满足子任务 3 的特殊限制。

【测试点约束】

对于所有测试点，满足 $1 \leq n, m \leq 2 \times 10^5$ ， $1 \leq h_i \leq 10^9$ 。
每个子任务的具体限制见下表：

子任务编号	分值	特殊限制
1	5	$n, m \leq 10$
2	15	每次给出的询问区间长度之和不超过 10^6
3	30	对于所有询问保证 $l = 1$
4	50	无

吞天得手 (ttds)

【题目背景】

平吞六国更何求，漏船载酒泛中流。得成蝴蝶寻花树，手把芙蓉朝九州。

【题目描述】

2918 年，已经隐退的帝君杨吞天迫于武鸣国的军事压力，重掌大权，他接下来要做的就是重振军队击溃敌军。

杨吞天一共有 n 名士兵，他们每个人都有一个武力值。现在他们排成一列，从前至后第 i 个人的武力值为 a_i 。现在杨吞天可能会选出至少一个人出列组成一支队伍（他们的前后顺序不会发生改变）。

定义一个共有 p 人的队伍 s 的战斗为

$$s_1 b^{p-1} + s_2 b^{p-2} + \cdots + s_p.$$

其中 s_i 表示队伍中从前往后第 i 个人的武力值。

定义队伍 s 的凝聚力大于队伍 t 的凝聚力当且仅当队伍 s 和队伍 t 的从前往后第一个武力值不同的人 i 满足 $s_i < t_i$ 或者 s 是 t 的严格前缀。

比起高级士兵的强度，帝君更在乎底层士兵的凝聚力，所以帝君现在他所有可能选出来的队伍凝聚力前 k 大的队伍战斗力是多少。

你作为 RM 大陆上最强的参谋长，帝君想要你回答这个问题。

你只需要回答答案模 998244353 的值。

【输入格式】

从文件 `ttds.in` 中读入数据。

输入共两行。

第一行三个正整数 n, k, b 。

第二行 n 个整数 a_1, a_2, \dots, a_n 。

所有输入的意义与【题目描述】中相同。

【输出格式】

输出到文件 `ttds.out` 中。

输出共 k 行，第 i 行为凝聚力第 i 大的队伍的战斗力模 998244353 的值。

【样例 1 输入】

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

【样例 1 输出】

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

【样例 2】

见选手目录下的 *ttds/ttds2.in* 与 *ttds/ttds2.ans*。

【测试点约束】

对于所有测试点，满足 $1 \leq n, k, a_i \leq 10^5$ ， $1 \leq b \leq 10^6$ ， $k \leq 2^n - 1$ 。
每个子任务的具体限制见下表：

子任务编号	分值	特殊限制
1	20	$n \leq 20$
2	30	$a_i \leq 30$
3	20	$k \leq 20$
4	30	无

【后记】

帝君之寿三千霜。