

NOIP 2024 模拟赛

CDQZ

测试时间：2024.08.20

题目名称	暴龙	笛卡尔树	战争	次方数
题目类型	传统型	传统型	传统型	传统型
目录	tree	cartesian	war	pow
可执行文件名	tree	cartesian	war	pow
输入文件名	tree.in	cartesian.in	war.in	pow.in
输出文件名	tree.out	cartesian.out	war.out	pow.out
提交文件名	tree.cpp	cartesian.cpp	war.cpp	pow.cpp
时间限制	1.5 秒	1.0 秒	2.0 秒	0.5 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
测试点数目	4	4	20	3
测试点是否等分	否	否	是	否
编译选项	-O2 -std=c++14			

【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 题面中的多行样例，数据将带有行号，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. 统一评测时采用的机器配置为：12th Gen Intel(R) Core(TM) i7-12700 2.10 GHz，内存 16GB。上述时限以此配置为准。
15. 评测在 Windows 11 下进行，使用 LemonLime 进行评测。如果对此条以及 14 条中的机器配置有疑问，请及时询问。
16. 题目大致按照难度升序排序。
17. 为了评测的流畅性，请选手不要提交故意延缓程序运行时间的代码。
18. 宣传：希望大家以后给每个部分放上样例。

暴龙 (tree)

【题目背景】

Hanghang 红温了，然后他变异成了暴龙。

【题目描述】

机房里有 k 种暴龙。现在他们占据了一棵树上的所有的点。这棵树有 n 个点，边有边权。现在对于某一种暴龙，他们的红温值定义为任意两个这种暴龙的距离的最大值。如果这种暴龙的出现次数小于 2，则红温值为 0。考拉想要证明 Hanghang 红温了，因此他想要问你每种暴龙的红温值。

【输入格式】

从文件 `tree.in` 中读入数据。

第一行两个正整数 n, k ，表示树的节点数和暴龙的种类数。

第二行 n 个在 $[1, k]$ 间的正整数 p_i ，表示每个节点的暴龙的种类。

接下来 $n - 1$ 行，每行三个正整数 a, b, c ，表示 a, b 之间有一条边，边权为 c 。保证这些边构成一棵树。

【输出格式】

输出到文件 `tree.out` 中。

输出 k 行，每行一个数表示种类为 i 的暴龙的红温值。

【样例 1 输入】

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

【样例 1 输出】

1	1
2	2
3	3

【样例 2】

见选手目录下的 *tree/tree2.in* 与 *tree/tree2.ans*。
该样例满足子任务 1 的限制。

【样例 3】

见选手目录下的 *tree/tree3.in* 与 *tree/tree3.ans*。
该样例满足子任务 2 的限制。

【样例 4】

见选手目录下的 *tree/tree4.in* 与 *tree/tree4.ans*。
该样例满足子任务 3 的限制。

【样例 5】

见选手目录下的 *tree/tree5.in* 与 *tree/tree5.ans*。
该样例满足子任务 4 的限制。

【测试点约束】

本题使用捆绑测试。

子任务编号	n	分值
1	3000	5
2	10^5	10
3	3×10^5	10
4	5×10^5	75

对于所有数据， $1 \leq p_i \leq k \leq n \leq 5 \times 10^5, 1 \leq a, b \leq n, 1 \leq c \leq 10^9$ 。
本题输入输出量较大，请选手使用较快的 IO 方式。

笛卡尔树 (cartesian)

【题目描述】

对于一个排列，它的大根笛卡尔树可以由以下性质唯一定义：

- 树是一棵二叉树，且树中的每个节点代表原排列中的一个数。
- 树的中序遍历结果是原始排列。
- 任何非根节点的父节点代表的值都比节点本身代表的值大。

设对于排列 a 中的一个数 i ，它在大根笛卡尔树上的深度为 $dep_{i,a}$ 。现在给定一个长为 n 的排列 p 和 m 次操作，每次操作交换其中两个相邻的元素，设得到的排列为 q ，求 $\sum_{i=1}^n |dep_{i,p} - dep_{i,q}|$ 。每次操作结束后，使 $p := q$ 。也就是说，一次操作会影响后续的操作。

【输入格式】

从 `cartesian.in` 中读入数据。
第一行两个正整数 n, m ，表示排列的长度和操作的次数。
第二行 n 个数 a_1, a_2, \dots, a_n ，表示这个排列。
接下来 m 行，每行一个数 p_i ，表示交换 a_{p_i} 和 a_{p_i+1} 。

【输出格式】

输出到文件 `cartesian.out` 中。
对于每个操作，输出一行表示要求的答案。

【样例 1 输入】

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

12	3
----	---

【样例 1 输出】

1	0
2	0
3	1
4	0
5	1
6	1
7	1
8	1
9	3
10	2

【样例 2】

见选手目录下的 *cartesian/cartesian2.in* 与 *cartesian/cartesian2.ans*。
该样例满足子任务 1 的限制条件。

【样例 3】

见选手目录下的 *cartesian/cartesian3.in* 与 *cartesian/cartesian3.ans*。
该样例满足子任务 2 的限制条件。

【样例 4】

见选手目录下的 *cartesian/cartesian4.in* 与 *cartesian/cartesian4.ans*。
该样例满足子任务 3 的限制条件。

【样例 5】

见选手目录下的 *cartesian/cartesian5.in* 与 *cartesian/cartesian5.ans*。
该样例满足子任务 4 的限制条件。

【测试点约束】

本题使用捆绑测试。

子任务编号	特殊性质	分值
1	$n, m \leq 5000$	10
2	A	15
3	B	20
4	无	55

特殊性质 A: $2|m$ ，且对于 $2|i$ ， $p_i = p_{i-1}$ 。

特殊性质 B: 保证 a 是随机排列，且 p_i 在 $[1, n - 1]$ 中随机选取。

对于所有数据， $1 \leq n, m \leq 3 \times 10^5, 1 \leq a_i \leq n, 1 \leq p_i < n$ 。

战争（war）

【题目背景】

你猜猜为什么题目名字叫战争？

【题目描述】

给定一棵以 1 为根的含 n 个节点的树，所有边的初始权值为 0。

有 m 次操作，每次操作可以写为 $1\ x\ y$ 或者 $2\ i$ 的形式，前者表示 x 节点到 y 节点的路径上的所有边权值增加 1，后者表示撤销第 i 次操作（保证第 i 次操作为增加操作且尚未被撤销）。

接下来有 q 次询问，每次询问可以写为 $l\ r$ 的形式，表示询问满足第 l 次操作后到第 r 次操作后权值始终不为 0 的深度最大的边（定义根节点的深度为 0，边的深度定义为连接的两个点的深度最大值）。

【输入格式】

从文件 *war.in* 中读取数据。

第一行三个整数 n, m, q ，分别代表树的节点数，操作次数，询问次数。

接下来 $n - 1$ 行，每行两个整数 u, v ，代表树上存在一条从 u 节点到 v 节点的边，保证所有边形成一棵树。

接下来 m 行，每行首先一个整数 op ，如果 $op = 1$ 则紧接着有两个整数 x, y ，否则如果 $op = 2$ 则紧接着有一个整数 i ，代表一次操作。

接下来 q 行，每行两个整数 l, r ，代表一次询问。

【输出格式】

输出到文件 *war.out* 中。

输出 q 行，每行一个整数代表该次询问的答案。

【样例 1 输入】

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


8	1 3 4
9	2 1
10	2 3
11	1 3
12	2 5
13	3 3
14	1 2
15	4 5

【样例 1 输出】

1	1
2	2
3	2
4	1
5	2

【样例 1 解释】

每条边的深度及每个时刻的权值如下：

1－2：1 2 3 2 1，深度为 1。

2－3：0 1 2 2 1，深度为 2。

2－4：0 0 1 1 0，深度为 2。

1－5：0 1 1 1 1，深度为 1。

因此询问区间 [1, 3], [2, 5], [3, 3], [1, 2], [4, 5] 的答案分别为 1, 2, 2, 1, 2。

【样例 2 输入】

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

11	1 6 4
12	1 4 2
13	1 4 1
14	2 5
15	2 6
16	2 4
17	4 7
18	1 3
19	3 3
20	1 7
21	1 4
22	4 5
23	3 4
24	1 1
25	1 1
26	1 5

【样例 2 输出】

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

【样例 3】

见选手目录下的 *war/war3.in* 与 *war/war3.ans*。
该样例满足测试点 1 ~ 4 的限制。

【样例 4】

见选手目录下的 *war/war4.in* 与 *war/war4.ans*。
该样例满足测试点 5 ~ 10 的限制。

【样例 5】

见选手目录下的 *war/war5.in* 与 *war/war5.ans*。
该样例满足测试点 11 ~ 16 的限制。

【样例 6】

见选手目录下的 *war/war6.in* 与 *war/war6.ans*。
该样例满足测试点 17 ~ 20 的限制。

【测试点约束】

测试点编号	$n, m \leq$	$q \leq$	特殊性质
1 ~ 4	5×10^3	5×10^3	无
5 ~ 10	10^5	5×10^5	$x = 1$
11 ~ 16	5×10^4	2×10^5	无
17 ~ 20	10^5	5×10^5	无

对于 100% 的数据，满足 $1 \leq n, m \leq 10^5, 1 \leq q \leq 5 \times 10^5, 1 \leq u, v \leq n, op \in [1, 2], 1 \leq x, y \leq n, x \neq y, 1 \leq i \leq m, 1 \leq l \leq r \leq m$ 。

次方数 (pow)

【题目描述】

你有一个正整数序列 a_1, a_2, \dots, a_n 。请问有多少四元组 $(l_1, r_1, l_2, r_2) (l_1 \leq r_1 < l_2 \leq r_2)$ ，满足 $\prod_{i=l_1}^{r_1} a_i \times \prod_{i=l_2}^{r_2} a_i$ 是 k 次方数。

【输入格式】

从文件 *pow.in* 中读取数据。
第一行两个整数 n, k 。
第二行 n 个数，表示 a_1, a_2, \dots, a_n 。

【输出格式】

输出到文件 *pow.out* 中。
输出一个整数，表示答案。

【样例 1 输入】

```
1 5 2
2 12 8 12 16 8
```

【样例 1 输出】

```
1 9
```

【样例 2】

见选手目录下的 *pow/pow2.in* 与 *pow/pow2.ans*。
该样例满足子任务 1 的限制。

【样例 3】

见选手目录下的 *pow/pow3.in* 与 *pow/pow3.ans*。
该样例满足子任务 1 的限制。

【样例 4】

见选手目录下的 *pow/pow4.in* 与 *pow/pow4.ans*。
该样例满足子任务 2 的限制。

【样例 5】

见选手目录下的 *pow/pow5.in* 与 *pow/pow5.ans*。
该样例满足子任务 3 的限制。

【测试点约束】

本题使用捆绑测试。

子任务编号	特殊性质	分值
1	$n \leq 14, a_i \leq 20$	10
2	$a_i \leq 10^{12}$	25
3	无	65

对于所有数据， $1 \leq n \leq 500, 1 \leq k \leq 10^9, 1 \leq a_i \leq 10^{36}$ 。