

2024 省选模拟赛

时间：2024 年 2 月 28 日

题目名称	数组	虚树	下落的数字
题目类型	传统题	传统题	传统题
目录	array	tree	fall
可执行文件名	array	tree	fall
输入文件名	array.in	tree.in	fall.in
输出文件名	array.out	tree.out	fall.out
每个测试点时限	3 秒	1 秒	1 秒
内存限制	512 MB	512 MB	512 MB
子任务数目	5	6	5
测试点是否等分	否	否	否

提交源程序文件名

对于 C++ 语言	array.cpp	tree.cpp	fall.cpp
-----------	-----------	----------	----------

编译选项

对于 C++ 语言	-O2 -std=c++14
-----------	----------------

注意事项（请仔细阅读）

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C/C++ 中函数 `main()` 的返回值类型必须是 `int`，程序正常结束时的返回值必须是 0。
3. 提交的程序代码文件的放置位置请参考各省的具体要求。
4. 因违反以上三点而出现的错误或问题，申诉时一律不予受理。
5. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
6. 选手提交的程序源文件必须不大于 100KB。
7. 程序可使用的栈空间内存限制与题目的内存限制一致。
8. 只提供 Linux 格式附加样例文件。
9. 评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以此为准。

数组 (array)

【题目描述】

有一个数组 a 和一个集合 S ，初始每一个数 a_i 都有一个值 0 或 1 且 S 为空。一开始每个数是多少都是未知的。每次可能进行以下三种操作：

1. **MERGE** $k\ T$ ：给定集合 T ，保证 T 的大小为 k 且与 S 的交集为空。令 $x = \bigvee_{i \in T} a_i$ ，对于集合 T 里所有 i ，将 a_i 修改为 x 。
2. **GET** $x\ y$ ：保证 $x \notin S$ ，已知当前 $a_x = y$ ，且若 y 等于 1 则将 x 加入集合 S 。
3. **QUERY** p ：询问在满足当前已知的所有信息的情况下，命题 $\forall i \notin S, a_i = 0$ 是否一定为真。若为假，则询问序列 $p, p+1, p+2, \dots, n, 1, 2, \dots, p-1$ 中第一个可能使得 $i \notin S \wedge a_i \neq 0$ 的数 i 。

【输入格式】

从文件 `array.in` 中读入数据。

本题强制在线。

第一行三个正整数 n, m, t ，分别表示数组大小，操作次数和强制在线常数。

接下来是 m 行，每行先输入一个字符串 s ：

- 若 s 为 **MERGE**，表示是操作 1，接下来输入一个数 k 表示集合 T 的大小，后面紧跟 k 个数 x_i ，表示集合 T 内的元素。
- 若 s 为 **GET**，表示是操作 2，接下来输入两个数 x, y 。
- 若 s 为 **QUERY**，表示是操作 3，接下来输入一个数 x 。

需要注意的是，输入中每一个 x ，令上一轮的询问结果为 $lastans$ （若没有询问或上次询问命题一定为真则为 0，否则为输出中唯一的数），真实值为 $((x-1+t \times lastans) \bmod n) + 1$ 。

【输出格式】

输出到文件 `array.out` 中。

对于每次 3 操作，第一行一个字符串为 **YES** 或 **NO**，表示命题是否为真。若输出为 **NO** 则需要再输出一个数 i ， i 的定义如题目描述。

【样例 1 输入】

```
1 5 8 0
2 QUERY 1
3 MERGE 2 1 3
4 GET 1 0
5 MERGE 3 4 2 3
6 GET 5 1
7 QUERY 5
8 GET 4 0
9 QUERY 2
```

【样例 1 输出】

```
1 NO 1
2 NO 2
3 YES
```

【样例 1 解释】

样例解释：

第一次询问，每个数的值都不知道，所以第一个可能不确定的数是 1。

第二次询问，当前可能的情况有以下两种：

0 0 0 0 1, 0 1 1 1 1

而集合 S 内只有元素 5，所以命题为假，从 5 开始的序列第一个可能不为 0 的位置是 2。

第三次询问，当前序列只能是 **0 0 0 0 1**，此时集合 S 内有元素 5，命题为真。

【数据范围】

子任务编号	分值	特殊性质
1	10	$n \leq 10, m \leq 100$
2	20	$n, m \leq 2000$
3	20	$n, m \leq 100000, t = 0$
4	20	$t = 0$
5	30	无

对于全部的数据, $1 \leq k, x \leq n \leq 500000$, $1 \leq m \leq 1000000$, $1 \leq \sum k \leq 1000000$, $0 \leq t, y \leq 1$ 。保证任意时刻输入总能存在一种最初的数组 a 使得每个 2 操作符合真实情况。

虚树 (tree)

【题目描述】

给你一棵 n 个点的树, 以 1 号点为根。现在你要给每个点染上 $1 \sim m$ 中的一种颜色, 使得每一种颜色构成的虚树大小之和最大, 输出这个最大值。

换句话说, 定义颜色 i 构成的虚树大小为所有颜色 i 的点以及颜色 i 的点之间的 LCA (最近公共祖先) 所构成的点集的大小, 设这个大小为 $f(i)$ (如果没有颜色为 i 的点则 $f(i) = 0$), 你需要最大化的是 $\sum_{i=1}^m f(i)$ 。

【输入格式】

从文件 `tree.in` 中读入数据。

第一行两个整数 n, m , 分别表示树的节点数和颜色数。接下来 $n - 1$ 行, 每行两个正整数 x, y , 表示树上的一条边 (x, y) 。

【输出格式】

输出到文件 `tree.out` 中。

输出一行一个整数表示最大的虚树大小之和。

【样例 1 输入】

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

【样例 1 输出】

```
1 7
```

【样例 1 解释】

一种最优的染色方案是将 1 染成第一种颜色, 2, 4 染成第二种颜色, 3, 5 染成第三种颜色。

【数据范围】

对于 100% 的数据, $1 \leq n \leq 2 \times 10^5, 1 \leq m \leq n$ 。

子任务编号	子任务分值	特殊限制
1	10	$n, m \leq 5$
2	5	给出的边均有 $y = x + 1$
3	15	给出的边均有 $x = 1$
4	20	$n \leq 5000, m \leq 20$
5	40	$n = m$
6	10	无

下落的数字 (fall)

【题目描述】

一棵以 1 为根的树, 每个点有一个权值, 一个数字从根节点出发, 不断按照如下策略行动:

- 在当前点的所有儿子中, 选择权值大于等于它且权值最小的儿子, 走到那个儿子上。
- 如果不存在这样的儿子, 它将停在当前节点。

现在有 m 次操作, 每次操作为修改一个点的权值, 或者查询一个数字 c 从根节点出发最终到达的点的编号。

【输入格式】

从文件 *fall.in* 中读入数据。

第一行两个数 n, m , 为树的节点数和操作数。

接下来一行 n 个整数, 第 i 个数表示 i 号点的权值 w_i 。

接下来 $n - 1$ 行, 每行描述树上的一条边 (x, y) 。

接下来 m 行每行描述一个操作, 有如下两种格式:

- 1 $a\ b$, 表示把 a 节点的权值改成 b 。
- 2 c , 表示查询数字 c 从根节点出发最终到达的点的编号。

注意 1 号点的权值并无意义, 你不需要关心它以及对它的修改 (尽管这样的修改可能存在)。数据保证任意时刻所有点的权值互不相同。

【输出格式】

输出到文件 *fall.out* 中。

对于每个查询操作, 输出一行一个整数, 表示数字 c 最终到达的节点的编号。

【样例 1 输入】

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

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

【样例 1 输出】

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

【数据范围】

对于 100% 的数据, $1 \leq n, m \leq 2 \times 10^5, 1 \leq w_i, b, c \leq 10^9, 1 \leq a_i \leq n$ 。

子任务编号	子任务分值	特殊限制
1	20	$n, m \leq 5000$
2	5	给出的边均有 $y = x + 1$
3	15	给出的边均有 $x = 1$
4	20	$m = 0$
5	40	无