最长待机（sleep）

【题目描述】

精灵程序员小 ω 和小 $\aleph$ 拥有无限的寿命，因此在写代码之余，它们经常玩一些对抗游戏来打发时间。尽管如此，时间还是太多，于是它们发明了一款专用于消磨时间的游戏：最长待机。

为了了解最长待机的规则，首先要了解精灵们使用的编程语言 Sleep++ 的规则：

- 程序由 n 个函数组成，第 $i(1 \le i \le n)$ 个函数具有种类 e_i 和子函数编号序列 $Q_i = (Q_{i,1}, Q_{i,2}, \cdots, Q_{i,l_i})$。$Q_i$ 可以为空，此时 l_i 为 0。
- n 以及所有的 e_i 和 Q_i 可以由程序员任意给出，但它们需要满足以下所有条件：
  - $n \ge 1$；
  - $\forall 1 \le i \le n$，$e_i \in \{0, 1\}$；
  - $\forall 1 \le i \le n$，Q_i 中元素两两不同且均为 $[i+1, n]$ 中的整数；
  - $\forall 2 \le j \le n$，恰好有一个 $Q_i(1 \le i \le n)$ 包含了 j。
- 调用函数 $i(1 \le i \le n)$ 时，按顺序执行如下操作：
  - 若 $e_i = 0$，令变量 r_i 为 1；否则程序员需要立即为 r_i 输入一个正整数值。
  - 若 Q_i 为空，程序等待 r_i 秒；否则重复以下操作 r_i 次：
    - 按顺序调用编号为 $Q_{i,1}, Q_{i,2}, \cdots, Q_{i,l_i}$ 的函数。
- 若一个种类为 1 的函数 j 被调用多次，则其每次调用都需要输入 r_j。
- 我们认为，在函数调用中，除了“等待 r 秒”之外的操作不消耗任何时间，即函数调用、运行和输入都在瞬间完成。因此，一个时刻内程序员可能输入多个数。

可以证明，调用任意一个 Sleep++ 程序的任意一个函数，无论如何设定输入，消耗的时间总是有限的。

“最长待机”的游戏规则如下：

- 小 ω 和小 $\aleph$ 准备好各自的 Sleep++ 程序并选择各自程序中的一个函数。它们互相知晓对方程序的结构以及选择的函数。
- 在时刻 0，小 ω 和小 $\aleph$ 同时调用自己选择的函数，游戏开始。
- 在时刻 t（$t \ge 0$），双方可以看到对方在时刻 0 至 $(t-1)$ 输入的所有数字，并相应调整自己在时刻 t 输入的数字，但双方无法得知对方在时刻 t 输入的数字。
- 函数调用先结束的一方输掉游戏，另一方胜利。两个调用同时结束算作平局。

小 ω 和小 $\aleph$ 都是绝顶聪明的，在它们眼中，如果有一方存在必胜策略，那么这局游戏是不公平的。换言之，双方都不存在必胜策略的游戏是公平的。

小 ω 写了一个 n 个函数的 Sleep++ 程序并进行了 m 次操作，操作有以下两种：

- 操作一：给出 k，将 e_k 修改为 $(1 - e_k)$；
- 操作二：给出 k，与小 $\aleph$ 玩一局“最长待机”，开始时小 ω 会调用自己的函数 k。

小 ℵ 信奉极简主义，它希望对于每一局游戏设计出函数个数最少的程序，使得选择其中某个函数能让这局游戏是公平的。你能帮它求出最少所需的函数个数吗？

可以证明，小 ℵ 总是能设计一个程序并选择其中一个函数，使得游戏是公平的。

【输入格式】

从文件 ***sleep.in*** 中读入数据。

输入的第一行包含两个正整数 n, m，表示小 ω 的程序中函数的个数以及操作次数。

接下来 n 行，第 i 行若干个整数，描述小 ω 程序中的函数 i：

- 前两个整数 e_i, l_i 表示函数种类和子函数编号序列长度；
- 接下来 l_i 个整数 $Q_{i,1}, Q_{i,2}, \cdots, Q_{i,l_i}$ 描述子函数编号序列。

接下来 m 行，第 j 行两个整数 o_j, k_j 描述一次操作，其中 $o_j = 1$ 表示操作一，$o_j = 2$ 表示操作二。

【输出格式】

输出到文件 ***sleep.out*** 中。

对于每个操作二输出一行一个整数，表示小 ℵ 的程序中最少所需的函数个数。

【样例 1 输入】

```
3 6
0 2 2 3
0 0
0 0
2 1
1 3
2 1
1 3
1 2
2 1
```

【样例 1 输出】

```
3
3
1
```

【样例 1 解释】

- 对于前两次游戏，小 ℵ 可以给出与小 ω 完全一致的程序并在游戏开始时调用函数 1。可以证明不存在函数个数更少的方案。
- 对于第三次游戏，小 ℵ 可以给出一个仅包含一个种类为 1 的函数的程序，并在游戏开始时调用函数 1。
  - 在时刻 0，小 ω 输入其程序中的 r_2，小 ℵ 输入其程序中的 r_1。
    - 注意：r 变量在小 ω 和小 ℵ 的程序之间是独立的，不会互相影响。
  - 输入完成后，小 ω 的程序在时刻 (r_2+1) 结束，小 ℵ 的程序在时刻 r_1 结束。
  - 由于两人在时刻 0 互不知道对方的决策，不能保证 (r_2+1) 和 r_1 的大小关系，故双方均不存在必胜策略，这局游戏是公平的。

【样例 2】

见选手目录下的 ***sleep/sleep2.in*** 与 ***sleep/sleep2.ans***。
该组数据满足特殊性质 AD。

【样例 3】

见选手目录下的 ***sleep/sleep3.in*** 与 ***sleep/sleep3.ans***。
该组数据满足特殊性质 BD。

【样例 4】

见选手目录下的 ***sleep/sleep4.in*** 与 ***sleep/sleep4.ans***。
该组数据满足特殊性质 D。

【样例 5】

见选手目录下的 ***sleep/sleep5.in*** 与 ***sleep/sleep5.ans***。
该组数据满足特殊性质 C。

【子任务】

对于所有测试数据，

- $1 \le n \le 5 \times 10^5$，$1 \le m \le 2 \times 10^5$；
- $\forall 1 \le i \le n$，$e_i \in \{0,1\}$，$0 \le l_i < n$；
- $\forall 1 \le i \le n, 1 \le j \le l_i$，$i < Q_{i,j} \le n$；
- $\forall 1 \le i \le n, 1 \le p < q \le l_i$，$Q_{i,p} \ne Q_{i,q}$；
- $\forall 2 \le j \le n$，恰好有一个 $Q_i (1 \le i \le n)$ 包含了 j；

- $\forall 1 \le j \le m$，$1 \le o_j \le 2$，$1 \le k_j \le n$。

测试点编号	$n \le$	$m \le$	特殊性质
$1 \sim 2$	3	24	无
3	80	400	AD
4			BD
$5 \sim 6$			D
7	3×10^5	10^5	AD
8			BD
$9 \sim 10$			D
11			A
12			BC
13			B
$14 \sim 15$			C
$16 \sim 17$			无
$18 \sim 19$	5×10^5	2×10^5	A
20			BC
21			B
$22 \sim 23$			C
$24 \sim 25$			无

特殊性质 A：保证

- 任意时刻 e_1 均为 0；
- $\forall 2 \le i \le n$，$l_i \le 1$；
- 操作二的 k 均为 1。

特殊性质 B：保证

- 操作二的 k 满足当时的 e_k 为 1。

特殊性质 C：保证

- $\forall 2 \le i \le n$，$i \in Q_{\lfloor \frac{i}{2} \rfloor}$；
- $\forall 1 \le i \le n$，序列 Q_i 单调递增。

特殊性质 D：保证

- 操作二不超过 10 个；
- 操作二的 k 均为 1。