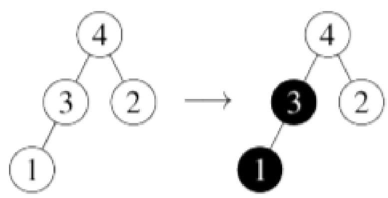
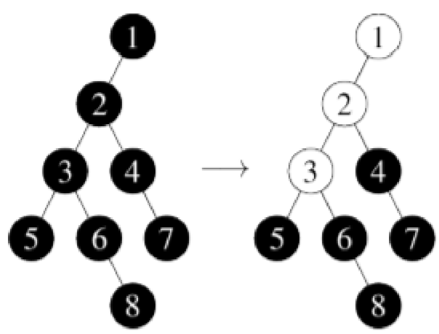
Boulder dash

It's 2021 and Jindiana Stones has found yet another dungeon to explore. The dungeon can be represented by a rooted tree with N rooms, numbered 1 through N, connected by $N-1$ corridors. Each room is either empty, or contains one boulder. The rooms are all initially empty. Q booby traps are triggered in order. These booby traps trigger one of the two following kinds of events.

- 1 x: x boulders appear *one by one* in room r and as long as a boulder has an empty room directly beneath it (a child in the tree representation) that is connected to the room it is currently in, it will roll down into it. If there are multiple such empty rooms beneath it, it will choose the room which has the least numbered room occurring in its subtree. If the boulder rolls down multiple rooms, it will make this decision at each level. Consider the tree below, if two boulders appear at the root (room 4), the first one goes from room 4 to room 3, since it contains a room numbered 1 in its subtree, then goes to room 1. The second boulder goes from room 4 to room 3 and

  stops there.
- 2 v: the boulder in room numbered v crumbles to dust and the room becomes empty. If any room in the tree is empty, but the room directly above it (its parent in the tree representation) has a boulder, then this boulder will roll down into it and the room directly above it will become empty. This process repeats till there is no such empty room in the tree. Consider the tree below, if boulders in rooms 5, 7 and 8 crumble to dust (in order), then rooms 1, 2 and 3 will become empty.

Input Format:

- The first line contains integers N and Q, the number of rooms, and the number of booby traps.
- The next N lines each contain an integer p_i, denoting that the room numbered p_i is a parent of the room numbered i in the rooted tree representation.
- The next Q lines contain two integers t and x if $t = 1$ or t and v if $t = 2$, where t denotes the type of booby trap, x denotes the number of boulders in the booby trap of the first kind and v denotes the room number in the booby trap of the second kind.

Output:

For each booby trap of the first kind, output the number of the room the last inserted boulder ends up in. For each booby trap of the second kind, output the number of boulders that roll down after the boulder from room v crumbles to dust.

Constraints:

- $1 \leq N, Q \leq 10^5$
- $1 \leq p_i \leq N$ or $p_i = 0$ if i is the root.
- It is guaranteed that there will never be a booby trap of the first kind which adds more boulders than there are empty rooms.
- It is also guaranteed that in every booby trap of the second kind, a boulder exists in node v.

Subtasks

- **Subtask 1 (10 points)**: The input forms a perfect binary tree. Each room has either 0 or 2 rooms directly beneath it, and all rooms with no rooms beneath them have the same distance from the root.
- **Subtask 2 (30 points)**: It is guaranteed that no boulders will roll down after a booby trap of the second kind.
- **Subtask 3 (30 points)**: It is guaranteed that there is exactly one booby trap of the first kind, and it is the very first booby trap.
- **Subtask 4 (30 points)**: original constraints.

Sample Input:

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

Sample Output: 1 3 2 2