Given a rooted binary tree, return the lowest common ancestor of its deepest leaves.

Recall that:

* The node of a binary tree is a *leaf* if and only if it has no children
* The *depth* of the root of the tree is 0, and if the depth of a node is d, the depth of each of its children is d+1.
* The *lowest common ancestor* of a set S of nodes is the node A with the largest depth such that every node in S is in the subtree with root A.

**Example 1:**

**Input:** root = [1,2,3]

**Output:** [1,2,3]

**Explanation:**

The deepest leaves are the nodes with values 2 and 3.

The lowest common ancestor of these leaves is the node with value 1.

The answer returned is a TreeNode object (not an array) with serialization "[1,2,3]".

**Example 2:**

**Input:** root = [1,2,3,4]

**Output:** [4]

**Example 3:**

**Input:** root = [1,2,3,4,5]

**Output:** [2,4,5]

**Constraints:**

* The given tree will have between 1 and 1000 nodes.
* Each node of the tree will have a distinct value between 1 and 1000.