Problem Link:

<https://leetcode.com/problems/lowest-common-ancestor-of-deepest-leaves/?envType=daily-question&envId=2025-04-04>

Solution:

/\*\*

\* Definition for a binary tree node.

\* struct TreeNode {

\* int val;

\* TreeNode \*left;

\* TreeNode \*right;

\* TreeNode() : val(0), left(nullptr), right(nullptr) {}

\* TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}

\* TreeNode(int x, TreeNode \*left, TreeNode \*right) : val(x), left(left), right(right) {}

\* };

\*/

class Solution {

public:

TreeNode\* lcaDeepestLeaves(TreeNode\* root) {

return dfs(root).first;

}

pair<TreeNode\*, int> dfs(TreeNode\* node) {

if(!node)

return {nullptr, -1};

auto left = dfs(node->left);

auto right = dfs(node->right);

if(left.second == right.second)

{

return {node, left.second + 1};

}

if(left.second > right.second)

{

return {left.first, left.second + 1};

}

return {right.first, right.second + 1};

}

};