Lowest Common Ancestor of Two Given Nodes

LeetCode

Given a binary tree, find the lowest common ancestor (LCA) of two given nodes in the tree.

“The lowest common ancestor is defined between two nodes p and q as the lowest node in T that has both p and q as descendants (where we allow **a node to be a descendant of itself**).”

Example:

```
               5
           /     \
         3         7
       /    \        \
     11       1        6
   /    \            /    \
 9        12       13       15
```

The First TargetNode = 1, Second TargetNode = 13

Output: The Lowest Common Ancestor: 5

Approach 1: Function to find the lowest common ancestor (LCA) of two nodes in a binary tree.

- If the root is null, return null (base case).
- If the current root is either **p** or **q**, it is the LCA.
- Recursively find the LCA in the left and right subtrees.
- If both left and right subtrees return non-null values, the current root is the LCA.
- If either left or right result is non-null, return the non-null value.
- **Time Complexity: O(N), where N is the number of nodes in the tree (each node is visited once).**
- **Space Complexity: O(H), where H is the height of the tree (recursion stack space).**