Lowest Common Ancestor of a Binary Tree

For this problem we use DFS recursion with the following idea:

Key Insight:

- If the current node is null, return null.
- If the current node is p or q, return it (a node is an ancestor of itself).
- Recursively search left and right subtrees:
  - If both left and right return non-null → current node is the LCA.
  - If only one returns non-null → propagate that node upward.

Algorithm:

1. Start from root.
2. Recursively check left and right subtrees.
3. Combine results:
   - Both sides non-null -> return current node.
   - Else return the non-null child.

Complexity:

- Time: O(n) - visit each node once.
- Space: O(h) - recursion stack (h = tree height).