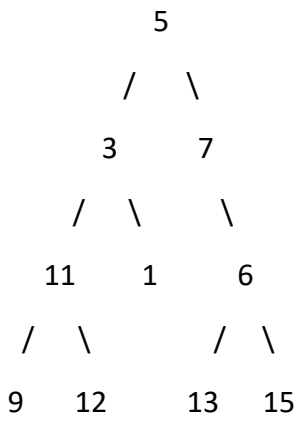
Boundary Traversal of Binary Tree GFG

Given a Binary Tree, find its Boundary Traversal. The traversal should be in the following order:

1. **Left boundary nodes:** defined as the path from the root to the left-most node ie- the leaf node you could reach when you always travel preferring the left subtree over the right subtree.
2. **Leaf nodes:** All the leaf nodes except for the ones that are part of left or right boundary.
3. **Reverse right boundary nodes:** defined as the path from the right-most node to the root. The right-most node is the leaf node you could reach when you always travel preferring the right subtree over the left subtree. Exclude the root from this as it was already included in the traversal of left boundary nodes.

Example:



Output: The Boundary Traversal of Binary Tree: 5, 3, 11, 9, 12, 1, 13, 15, 6, 7

Approach 1: Recursive function to collect the boundary nodes of the binary tree. (Traverse Left Nodes (excluding leaf nodes), Traverse Leaf Nodes, Traverse Right Nodes in reverse (excluding leaf nodes)

- The recursive approach collects boundary nodes by traversing the left boundary, collecting leaf nodes, and then traversing the right boundary.
- It maintains separate helper functions for each of these three tasks.
- The boundary traversal starts with adding the root to the result and then invoking the helper functions.

Time Complexity:

- **The time complexity is O(N), where N is the number of nodes in the tree, as each node is visited once.**

Space Complexity:

- **The space complexity is O(N) in the worst case, primarily due to the recursion stack.**