Binary Search Trees

- Insertion - A new key is always inserted at the leaf by maintaining the property of the binary search tree. We start searching for a key from the root until we hit a leaf node. Once a leaf node is found, the new node is added as a child of the leaf node. The below steps are followed while we try to insert a node into a binary search tree:
    - Initialize the current node (say, currNode or node) with root node
    - Compare the key with the current node.
    - Move left if the key is less than or equal to the current node value.
    - Move right if the key is greater than the current node value.
    - Repeat steps 2 and 3 until you reach a leaf node.
    - Attach the new key as a left or right child based on the comparison with the leaf node's value.
- Traversal
    - Preorder Traversal
        - The root node of the subtree is visited first.
        - Then the left subtree is traversed.
        - At last, the right subtree is traversed.
    - Postorder Traversal
        - The root node of the subtree is visited first.
        - Then the left subtree is traversed.
        - At last, the right subtree is traversed.
    - Inorder Traversal
        - The left subtree is traversed first
        - Then the root node for that subtree is traversed
        - Finally, the right subtree is traversed
    - Level Order Traversal
        - Given the root of a binary tree, return the level order traversal of its nodes' values. (i.e., from left to right, level by level).