**Largest BST in binary tree**

1st approach bruteforce.

1) start with the root node and try to check if it is the valid BST. If it is then send this root node to other method which will give the count of the node in that BST.

Do this for all the node and return the max of it.

2) start from the leaf node….single node is always a BST. Verify the condition left.maxNode < root.data && root.data < right.minNode for every node. If it satisfies then increase the BST size by 1

**Recover BST is 2 nodes are swapped**

1st approach bruteforce

1) traverse BST in any order and sort it.

2) again do the inorder traversal of the BST, and compare the node data with the array data. If it is not same then just then make array data as root data.

1) check if there is one violation or 2 violation…..if there is 2 violation then swap first and last violation

2) if there is only one violation then swap adj node of the violated node.