Trees Assignment **Batch: Interview Preparation**

1. Implement Binary Search tree class. It should have the following functions.
    a. insert
    b. remove
    c. search
    d. size
    e. isEmpty
2. Given a binary tree test if its a BST
3. Create a BST from a sorted array
4. Convert a BST into a sorted Linked List
5. You have a binary tree with non-negative numeric values stored in its nodes, you need to find out the maximum possible sum of all the nodes. Selection of nodes for the sum follows following constraint: If you have selected any node for the sum, you can not select its immediate parent and its children for sum.
6. Given preorder and inorder create the tree, given postorder and inorder create the tree.
7. Given two trees, return true if they are structurally identical they are made of nodes with the same values arranged in the same way.
8. Preorder traversal iteratively.
9. Given a binary tree. Print the level order traversal, **make sure each level start at a new line.**
10. Given a binary tree print all nodes that don’t have a sibling.
11. Given a binary tree, check whether there are two nodes in it whose sum equals a given value.
12. Given a binary search tree and a int s, find pair of nodes in the BST which sum to s.
    a. Find a solution for which time complexity is O(n)
    b. Find a solution which uses maximum O(logn) extra space.
13. Find LCA of two elements in a BST.
14. Given a Binary tree find the largest BST subtree.

15. Given a tree and a node of tree, print all nodes which are at distance k from the given node.
16. Given a binary tree and a number k, print out all root to leaf paths where the sum of all nodes value is same as the given number.
17. Given a binary search tree and a int s, find pair of nodes in the BST which sum to s.
    a. Find a solution for which time complexity is O(n)
    b. Find a solution which uses maximum O(logn) extra space.