CL2001 – Data Structure Lab

Exercise # 09

Note:

- Copied task will be awarded **zero** marks.
- Use comments wherever applicable.
- Note that these lab task marks could be graded through a viva in lab.
- Variables and functions names should be meaningful.

Problem: 1 |

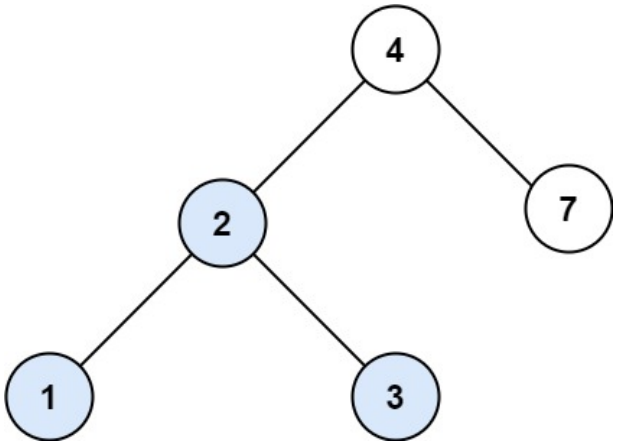
1. Write a function that performs Pre order Traversal of a BST.
2. Write a function to find the largest value in BST.
3. Write a function to count the leaf nodes in a BST.

Problem: 2 | Search in a Binary Search Tree

You are given the root of a binary search tree (BST) and an integer val.

Find the node in the BST that the node's value equals val and return the subtree rooted with that node. If such a node does not exist, return null.

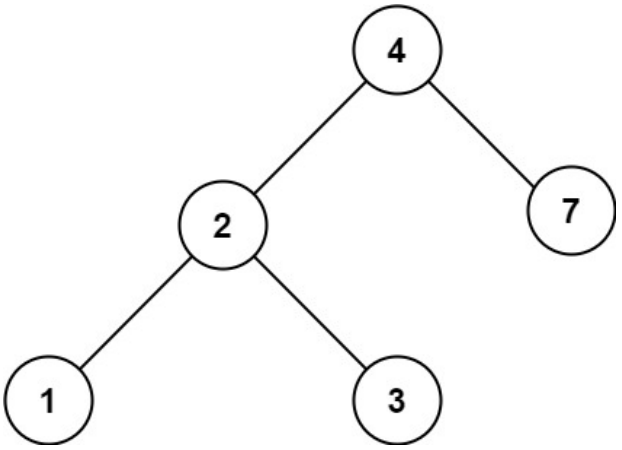
Example 1:



Input: root = [4,2,7,1,3], val = 2

Output: [2,1,3]

Example 2:



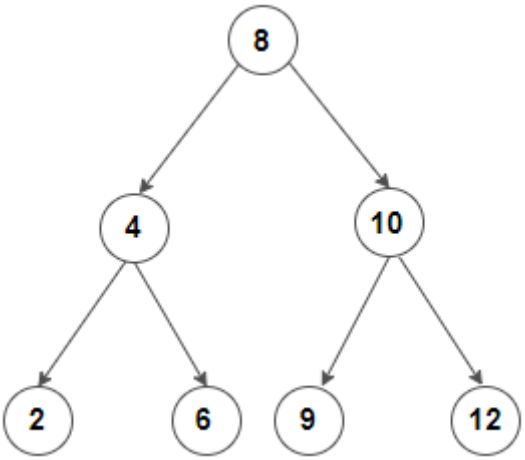
Input: root = [4,2,7,1,3], val = 5

Output: []

Problem: 3 | Floor and Ceil in a Binary Search Tree

Given a BST, find the floor and ceil of a given key in it. If the given key lies in the BST, then both floor and ceil are equal to that key; otherwise, the ceil is equal to the next greater key (if any) in the BST, and the floor is equal to the previous greater key (if any) in the BST.

For example, consider the following tree:



The floor of 3 is 2, ceil of 3 is 4

The floor of 9 is 9, ceil of 9 is 9

The floor of 7 is 6, ceil of 7 is 8

The floor of 14 is 12, ceil of 14 does not exist

The floor of 1 does not exist, ceil of 1 is 2

Hint:

Search for the given key in the tree and update the ceil to the current node before visiting its left subtree. Similarly, update the floor to the current node before visiting its right subtree. If the key is found in the BST, then the floor and ceil are equal to that key. If the key is not found in the BST, then the floor and ceil were already updated while searching for the key.