LAB 6 : Binary Search Tree

[CO2]

Instructions for students:

- Complete the following methods.
- You may use Java / Python to complete the tasks.
- DO NOT CREATE a separate folder for each task just follow the given template.
- If you are using **JAVA**, then follow the **Java Template**.
- If you are using **PYTHON**, then follow the **Python Template**.

NOTE:

- **YOU CANNOT USE ANY OTHER DATA STRUCTURE OTHER THAN ARRAY UNLESS MENTIONED IN THE QUESTION.**
- **YOUR CODE SHOULD WORK FOR ANY VALID INPUTS.**

Python List, Negative indexing and append() is STRICTLY prohibited unless mentioned in the question

Lab Tasks: 3 tasks (1~3)

Assignment Tasks: 3 tasks (4~6)

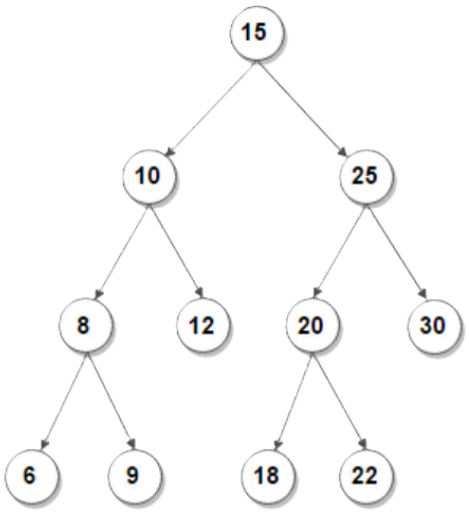
Total Assignment Mark: 3*5=15

LAB TASKS (no need to submit)

1. Find the Lowest Common Ancestor (LCA) of two nodes in a BST

The lowest common ancestor (LCA) of two nodes x and y in the BST is the lowest (i.e., deepest) node that has both x and y as descendants. In other words, the LCA of x and y is the shared ancestor of x and y that is located farthest from the root.

Corner Case: if y lies in the subtree rooted at node x, then x is the LCA; otherwise, if x lies in the subtree rooted at node y, then y is the LCA.



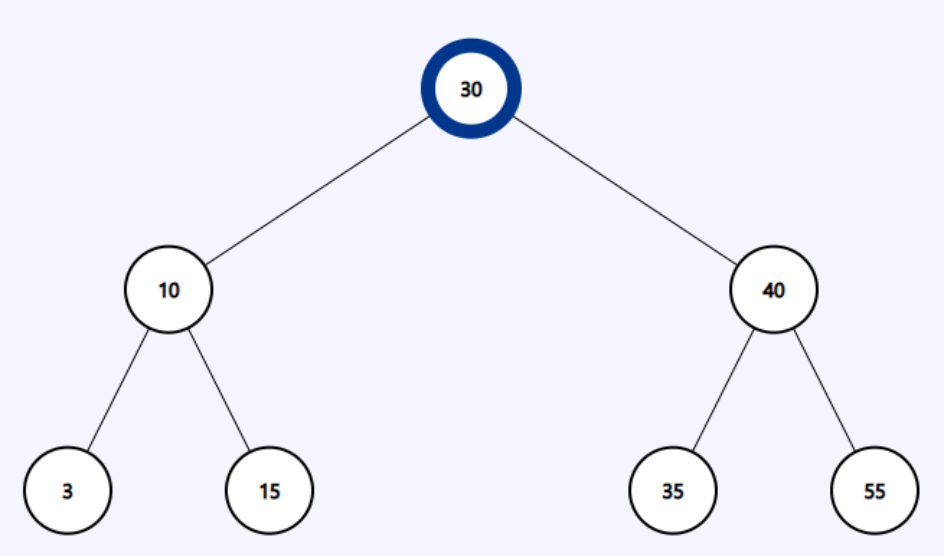
LCA(6,12) = 10
LCA(20,6) = 15
LCA(18,22) = 20
LCA(20,25) = 25
LCA(10,12) = 10

2. Find the path

Faria studies in Rangpur Medical College. During her Eid vacation, she is planning to visit her family and relatives in Chittagong. But she feels really bad during the journey. As a result, you need to help Faria reach her destination.

A Binary Search Tree is given.The root node is the starting position of Faria and a key node will be given as the destination. Now, you have to find if the following key node (which is the destination of Faria) is present in the tree or not. If present, return the path from the root node to the key node else print “No Path Found”.

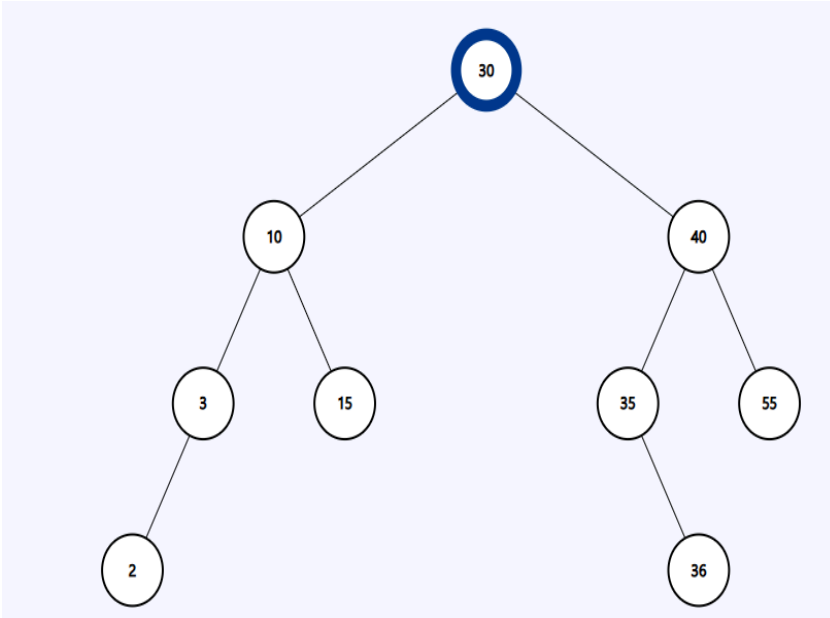
[You can use Python List for this task]



Sample Input	Sample Output
Source node(root): 30 Destination node(key): 15	Path: [30, 10, 15]
Source node(root): 30 Destination node(key): 50	No Path Found

3. Sum of all Leaf nodes of a BST

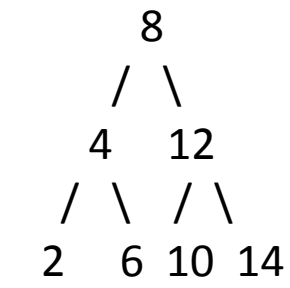
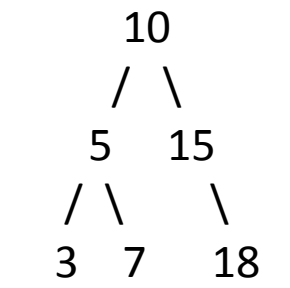
Write a function **sum_of_leaves(root)** that takes the root of a binary search tree as input and prints the sum of all the leaf nodes of the tree.

Sample Input	Sample Output
	Sum of all leaves=2+15+36+55=108

ASSIGNMENT TASKS (must submit)

4. Sum of range

Write a function **rangeSum(root,low,high)** that takes the root of a Binary Search Tree as a parameter and sums up the values within the range mentioned in the parameter.

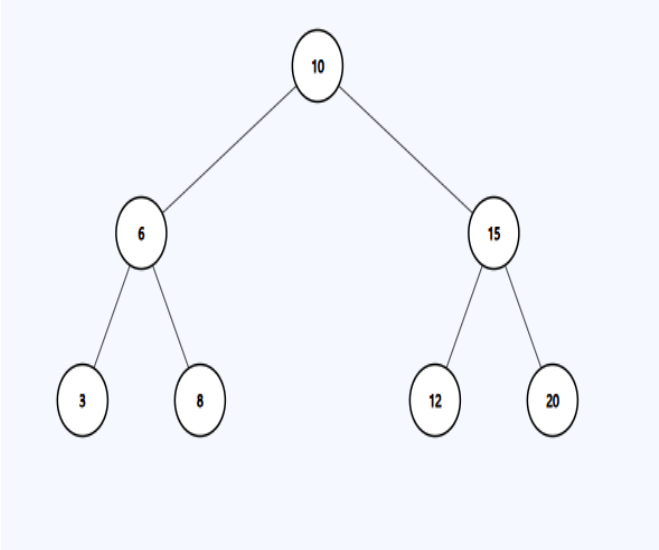
Sample Input	Sample Function Call	Sample Returned Value
	`rangeSum(root, 5, 13)`	**36** **Explanation:** 6+8+10+12=36
	`rangeSum(root, 7, 15)`	**32** **Explanation:** 7+10+15=32

5. Mirror Sum of a BST

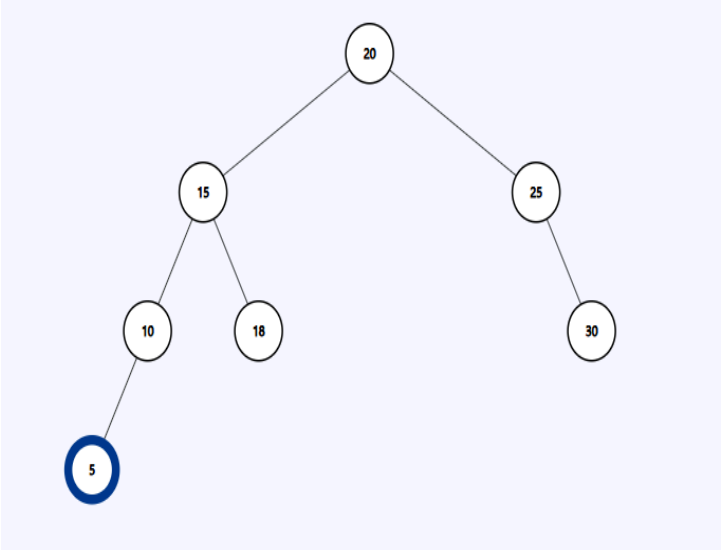
You are given the root node of a Binary Search Tree. You need to calculate the **sum** of the values of the nodes that are **mirrors** of each other then **return** it. Here, mirror means the nodes that are located in corresponding positions in the left and right subtrees of the root. If the mirror doesn't exist then do not sum.

Note: You can use helper functions.

Example Tree input 1



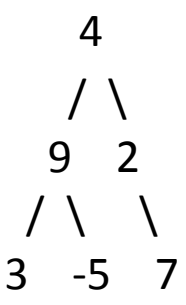
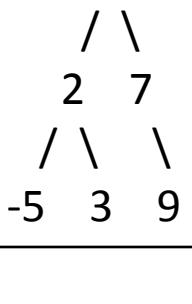
Example Tree input 2



Sample Input	Sample Output	Explanation
mirrorSum(root)	64	For Tree 1 Mirror nodes are: 6 and 15, sum = 6 + 15 = 21 3 and 20, sum = 3 + 20 = 23 8 and 12, sum = 8 + 12 = 20 Total Mirror Node Sum = 21+23+20 = 64
mirrorSum(root)	80	For Tree 2 Mirror nodes are: 15 and 25, sum = 15 + 25 = 40 10 and 30, sum = 10 +30 = 40 Total Mirror Node Sum = 40 + 40 = 80

6. Check BST

Write a function **isBST(root)** that takes the root of a Binary Tree as a parameter and then returns true if its a BST otherwise returns false.

Sample Input	Sample Returned Value
	**false**
	**true**