Start Time: 8:15 AM
Total Marks: 50   Completion Time: 11:15 AM

Objective:

In this lab, students will practice about the working of Binary Search Tree and basics of Heap Tree.

Instructions:

1) Follow the question instructions very carefully, no changes in function prototypes are allowed.
2) Your laptops must be on airplane mode.
3) Anyone caught in an act of plagiarism would be awarded an "F" grade in this Lab.

TASK-01: Find the Kth Ancestor in a BST   [20 Marks]

You are given a binary search tree and a target node within this tree. Your task is to find the Kth **ancestor** of the target node. The Kth ancestor of a node is defined as the ancestor that is K levels above the node.

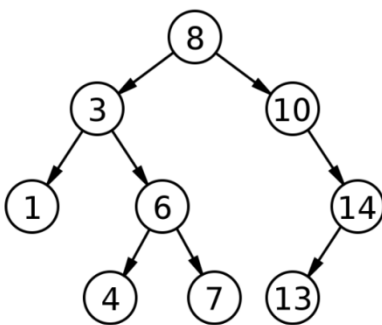If the target node does not have a Kth ancestor (K is larger than the depth of the node), return -1.

          Note: *No extra array or data structure should be used to solve this problem.*

 Function prototype:

                    int findKthAncestor(TreeNode* root, int target, int K)

Sample Run:

Input:

          target = 13, k = 2

Output: 10

TASK-02: Check if a BST is a Subtree of Another BST   [20 Marks]

You are given two Binary Search Trees, t1 and t2. Write a function to determine whether t2 is a **subtree** of t1.

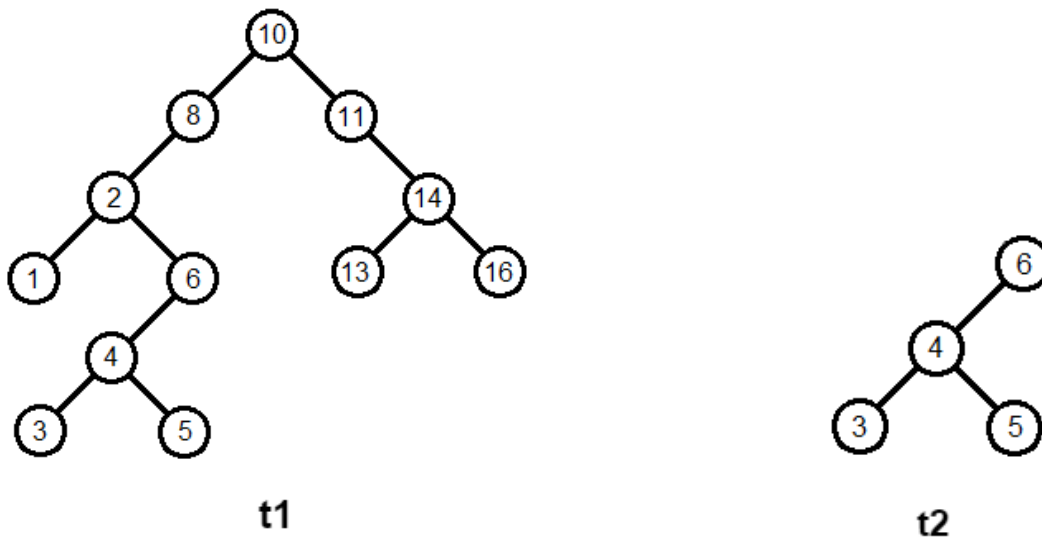A subtree of a tree is a tree that is a part of another tree.

Note: *You are not allowed to use additional data structures for storing nodes.*

Function prototype:

bool isSubtree(TreeNode* t1, TreeNode* t2)

Sample run
Input:

Output: true

TASK-03: Convert Min Heap to Max Heap   [10 Marks]

You are given a Min Heap, your task is to convert it to Max Heap.

Function prototype:

void convertToMaxHeap(int arr[], int n)

Sample Run:

Input: {3, 5, 9, 6, 8, 20, 10, 12, 18, 9}

Output: {20, 18, 10, 12, 9, 9, 3, 5, 6, 8}