Data Structures and Algorithms
CS-F22
LAB-07

Issue Date: August 16,2024

Start Time: 8:15 AM

Total Marks : 50

Completion Time: 11:15 M

Objective:

In this lab, students will understand about the working of binary tree.

Instructions:

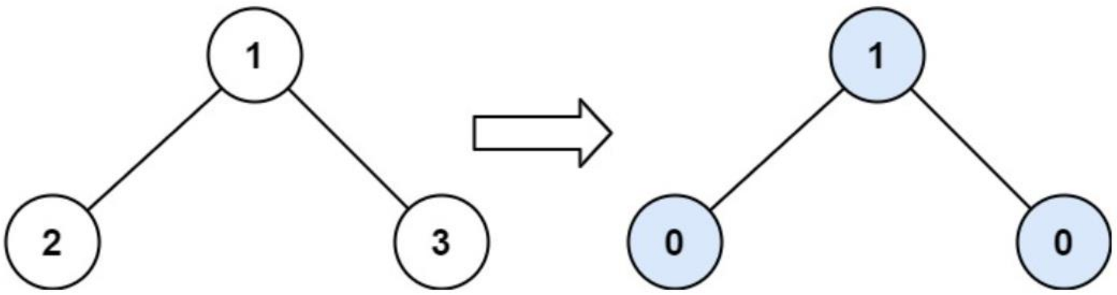
1) Follow the question instructions very carefully, no changes in function prototypes are allowed.
2) Your laptops must be on airplane mode.
3) Anyone caught in an act of plagiarism would be awarded an “F” grade in this Lab.

TASK 01: Binary Tree Tilt [10 Marks]

Given the root of a binary tree, return *the sum of every tree node's* ***tilt***.
The **tilt** of a tree node is the **absolute difference** between the sum of all left subtree node **values** and all right subtree node **values**. If a node does not have a left child, then the sum of the left subtree node **values** is treated as 0. The rule is similar if the node does not have a right child.

Function prototype : int findTilt(); //Member Function

Input: root = [1,2,3]
Output: 1
Explanation:



Tilt of node 2 : |0-0| = 0 (no children)
Tilt of node 3 : |0-0| = 0 (no children)
Tilt of node 1 : |2-3| = 1 (left subtree is just left child, so sum is 2; right subtree is just right child, so sum is 3)
Sum of every tilt : 0 + 0 + 1 = 1

TASK 02: Maximum Binary Tree [20 Marks]

You are given an integer array nums with no duplicates. A maximum binary tree can be built recursively from nums using the following algorithm:

1. Create a root node whose value is the maximum value in nums.
2. Recursively build the left subtree on the subarray prefix to the left of the maximum value.
3. Recursively build the right subtree on the subarray suffix to the right of the maximum value.

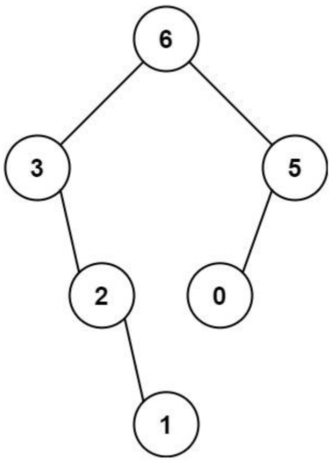
Return *the maximum binary tree built from* nums. Think about the prototype yourself.

Sample run:
Input: nums = [3,2,1,6,0,5]
Output: [6,3,5,-1,2,0,-1,-1,1]

//-1 indicating no child

Explanation: The recursive calls are as follow:

- The largest value in [3,2,1,6,0,5] is 6. Left prefix is [3,2,1] and right suffix is [0,5].
  - The largest value in [3,2,1] is 3. Left prefix is [] and right suffix is [2,1].
    - Empty array, so no child.
    - The largest value in [2,1] is 2. Left prefix is [] and right suffix is [1].
      - Empty array, so no child.
      - Only one element, so child is a node with value 1.
  - The largest value in [0,5] is 5. Left prefix is [0] and right suffix is [].
    - Only one element, so child is a node with value 0.
    - Empty array, so no child.



TASK 03: Display binary tree in parenthesized form [20 Marks]

The parenthesized view of a binary tree is a textual representation that shows the hierarchical structure of the tree using parentheses. This representation helps visualize the tree's structure in a compact format.

Root Node: The root node is displayed first.

Subtrees: Each subtree of the root is enclosed in parentheses. The left subtree appears immediately after the root node, followed by the right subtree.

Function prototype : void displayParenthesizedView(const int tree[], int size)

// -1 indicating no child

Sample run:

Input Tree: {1, 2, 3, -1, -1, 4, 5}

Output: 1(2,3(4,5))