# Algorithms Laboratory (CS29203)
# Assignment 5: Binary Search Trees
# Department of CSE, IIT Kharagpur

**26th September 2024**

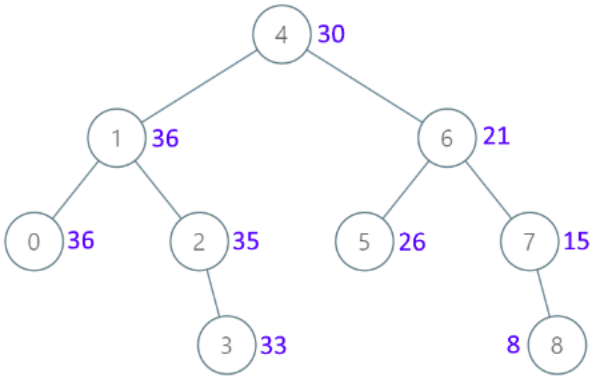
---

## Question-1

**(50 points)**
Consider a BST having integer data stored in its nodes. Let the data associated with $i$-th node is denoted as $d_i$. Let there are $n$ nodes in the tree and the set of all data is denoted as $\mathcal{D} = \{d_i\}_{i=1}^{n}$. Now our goal is to convert the BST into another tree where the data stored in each node of the tree will be changed as follows. The data stored in the $i$-th node is changed from $d_i$ to $d'_i$, where

$$d'_i = d_i + \sum_{j=1}^{n} d_j, \quad \text{such that} \quad d_j > d_i.$$

That is, the new value of a node will be the summation of its original value and all other nodes in the BST whose values are greater than that node. For example, consider the following BST.




The original data of each node are shown inside the nodes, and the new values are shown outside the nodes in blue color. Consider the root node, whose original value is 4. Other nodes having values stored $> 4$ are $5, 6, 7, 8$. Hence the new value of the root node will be $4 + 5 + 6 + 7 + 8 = 30$. This is applicable to all other nodes in the tree.

Given a BST, your task is to **convert** the BST into a new tree (which may not necessarily be a BST). **Do not** create a new tree, you have to **modify** the original tree. The output format of the nodes is to be defined in level order from left to right and a missing child is denoted as $-1$ (see the example below, and assume that there is no node in the tree will have the value of $-1$).

**Important:** You are **not** allowed to implement **array** based representation of trees (you have to use standard **linked** representation of trees).

Example:

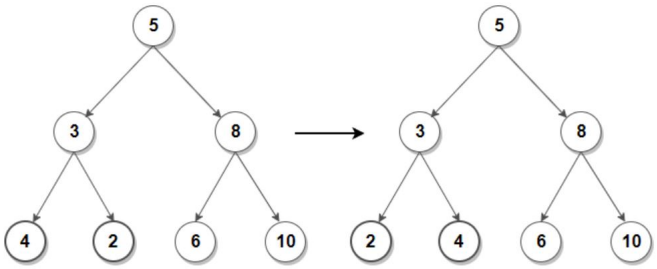
```
(Input) Enter the number of nodes in the tree: 9
        Enter the tree nodes [ ]: 4  1  6  0  2  5  7 3  8

(Output) Building the BST is done..
         Modified tree is: 30 36  21  36  35  26  15  -1  -1  -1  33  -1  -1  -1  8
```

# Question-2

**(50 points)**

Consider a binary tree where each node stores an integer number. The numbers in the tree are arranged in such a way that interchanging two particular nodes in the tree will result the tree to be a Binary Search Tree (BST). For example, consider the following binary tree in the left side. In this tree, interchanging nodes having values 2 and 4 will result the tree to become a BST, as shown in the right side.




Given a binary tree, your task is to **convert** this into a BST assuming that interchanging only two nodes will be sufficient for this purpose. Following are the sequence of tasks you have to perform. First you will create a BST. Print the inorder traversal of the tree, which will give you a sorted sequence of numbers.

Then interchange the values of any two nodes in the BST, which will result in making the tree into a non-BST. Write a function `interchangeNodes(node1, node2)` that will exchange the values stored at two nodes of the BST. You can pass the address of any two nodes of the BST in this function. For example, `node1` can be `root->left`, and `node2` can be `root->right->left->right`. Print the inorder traversal of the modified tree. The sequence will not be sorted at this stage.

Now you have to **convert** the modified tree into a BST. Write a function `restoreBST( )` to perform this task. **Do not** create a new tree, you have to **modify** the already created tree into a BST. Finally, perform an inorder traversal of the modified tree and print the values. The sequence should be sorted now.

**Important:** You are **not** allowed to implement **array** based representation of trees (you have to use standard **linked** representation of trees).

Example:

```
(Input) Enter the number of nodes in the tree: 7
        Enter the value of the nodes: 15  10  20  8  12  16  25

(Output) Building the BST is done..
         Inorder traversal of the BST is: 8  10  12  15  16  20  25

        Interchanging the values of two nodes..
        Inorder traversal of the modified tree is: 8  12  10  15  16  20  25

        Rebuilding the tree is done..
        Inorder traversal of the modified tree is: 8  10  12  15  16  20  25
```