National University of Computer & Emerging Sciences (NUCES) Islamabad,
Department of Computer Science

DATA STRUCTURES – FALL 2021

LAB 10

Learning Outcomes

In this laboratory, you will implement operations of Binary Search Tree.

National University of Computer & Emerging Sciences (NUCES) Islamabad,
Department of Computer Science

TASK 1

We have discussed the structure and how to access each node of BST in the previous lab. In this lab we will perform deletion and retrieval in BST.

Note: You are allowed to use the implementation from the previous lab and add functionality to that code.

Implement the following operations of BST.

BSTree ()
Constructor. Creates an empty binary search tree.

~BSTree ()
Destructor. Deallocates (frees) the memory used to store a binary search tree.

insert ()
Inserts new DataItem into a BST. If a data item with the same key as newDataItem already exists in the tree, then updates that data item's nonkey fields with newDataItem's nonkey fields.

retrieve ()
Searches BST for the data item with the user given key. If this data item is found, it returns true. Otherwise returns false.

delete()
Search for the data item with the user given key and delete it. Remember deletion of a node can follow three cases.

- Deletion of leaf node
- Deletion of node having one child
- Deletion of node having two children

Make sure you handle all the cases.

displayPreOrder()
Displays the dataItem of nodes in pre-order manner.

displayPostOrder()
Displays the dataItem of nodes in post-order manner.

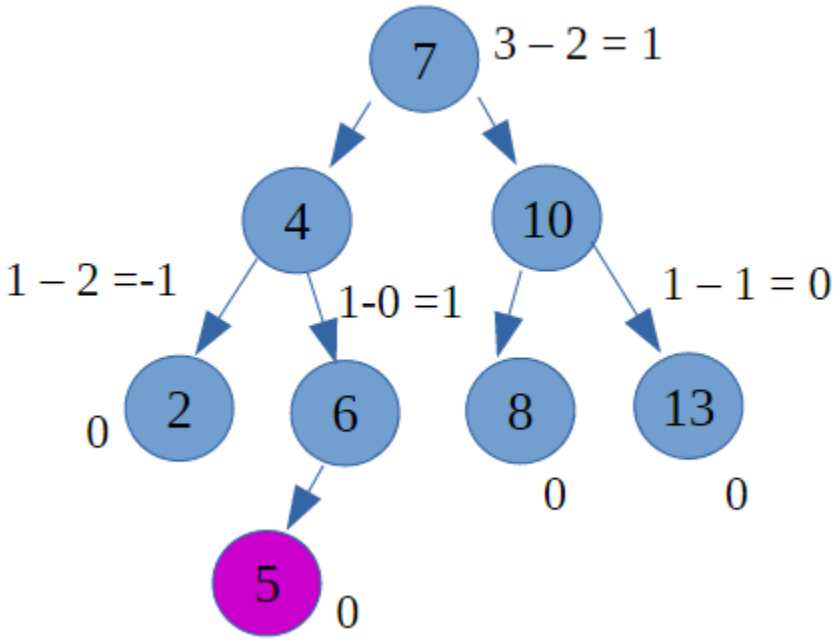
TASK 2

Implement a function **height()** that finds the height i.e. the maximum depth of a BST.

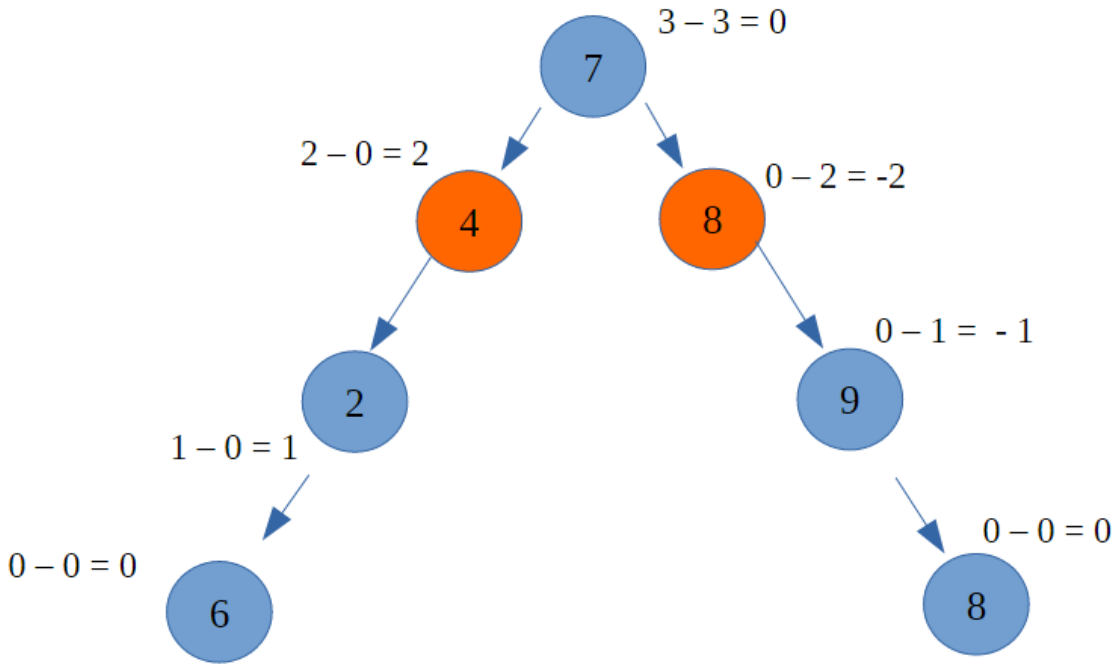
BONUS TASK

Create a function **isBalanced()** to determine whether the tree is height balanced or not.

A tree is balanced when the difference between the heights of subtrees under each node is not greater than one. Consider the figure below which shows a balanced tree.



The tree given below is not a balanced tree.


3 – 3 = 0
7
2 – 0 = 2
4
0 – 2 = -2
8
0 – 1 = - 1
2
9
1 – 0 = 1
0 – 0 = 0
6
0 – 0 = 0
8