Data Structures and Algorithms

Comp 200

Fall 2022

Department of Computer Science

Forman Christian College University

Lab 9
Tree - Binary Tree

In Lab Problems

Question 1:

Consider the following Node class:

```
class Node:

     def __init__(self,data,parent=None,left=None,right=None):

          self.data = data

          self.parent = parent

          self.left = left

          self.right = right
```

Implement the BinaryTree class as we discussed in the lecture.

```
class BinaryTree:

     def __init__(self):

     self.root = None

     self.size = 0
```

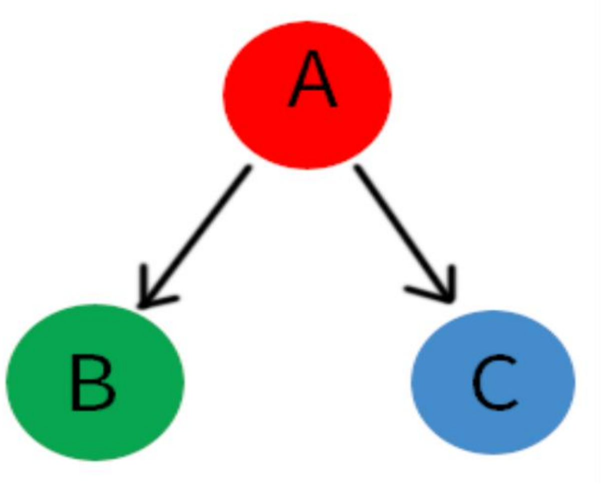
Other methods are given below:

id	*Method*	*Comments*
1	`__init__(self)`	Initialize the tree (already implemented).
2	`AddLeft(self,node,data)`	Adds a new node to the left of the given node ‘node’
3	`GetLeft(self,node)`	Returns the left child of the given node ‘node’
4	`AddRight(self,node,data)`	Adds a new node to the right of the given node ‘node’
5	`GetRight(self,node)`	Returns the right child of the given node ‘node’
6	`SetRoot(self, data)`	Adds a new root node
7	`GetRoot(self)`	Returns reference to the root node
8	`is_Leaf(self, node)`	Checks if a given node is a leaf node
9	`GetParent(self, node)`	Gets the parent of ‘node’
10	`__len__(self)`	Returns the size of the Tree

Question 2

Manual Tree Creation

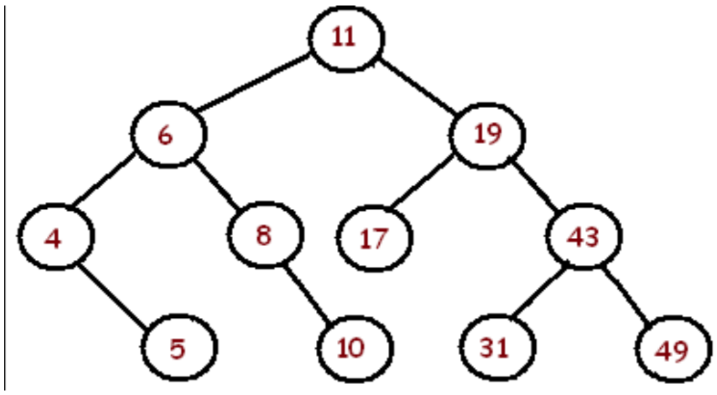
Consider the following simple tree (with ‘A’ as the root):



It can easily be constructed in python using the following code:

```
T = BinaryTree()

T.SetRoot( Node('A') )

root = T.GetRoot()

T.AddLeft(root, Node('B'))

T.AddRight(root, Node('C'))
```

Using the same method, construct a tree as shown in the figure below:



Question 3

Write the code to display the above tree by using Post order, preorder, inorder and BFS traversal.