Course: Data Structures and Algorithms Assignment 4

Binary Tree *Dr. Belal Al-Fuhaidi*

1- Mark the following statements as true or false.

a. A binary tree must be nonempty.

صحيح

b. The level of the root node is 0.

صحيح

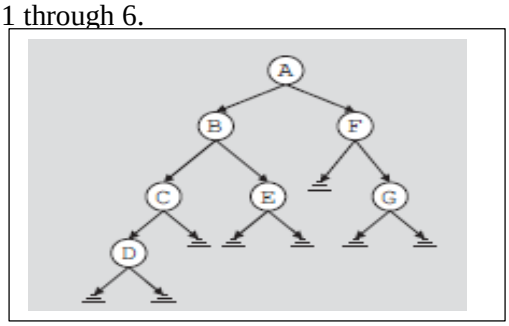
c. If a tree has only one node, the height of this tree is 0 because the number of levels is 0.

صحيح

d. The inorder traversal of a binary tree always outputs the data in ascending order.

خاطئ

2- The binary tree of the following Figure is to be used for Exercises 1 through 6.



1. Find L_A, the node in the left subtree of A.

L_(A) = D

2. Find R_A, the node in the right subtree of A.

R_(A) = E

3. Find R_B, the node in the right subtree of B.

R_(B) = G

4. List the nodes of this binary tree in an inorder sequence.

D, A, E, B, G, C, F

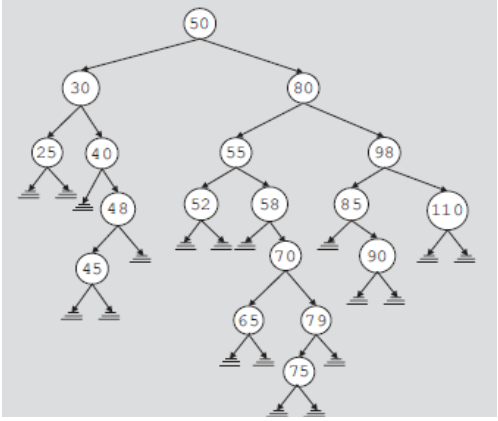
5. List the nodes of this binary tree in a preorder sequence.

A, D, B, E, C, G, F

6. List the nodes of this binary tree in a postorder sequence.

E, B, G, F, C, A

3- The binary search tree of the following Figure is to be used for Exercises 1through 4.



1. List the path from the node with info 80 to the node with info 79.

80, 65, 85,79

2. A node with info 35 is to be inserted in the tree. List the nodes that are visited by the function insert to insert 35. Redraw the tree after inserting 35.
3. Delete node 52 and redraw the binary tree.

بعد حذف ٥٢ ،
تظل الشجرة دون تغيير حيث أن ٥٢ هي عقدة ورقية

4. Delete node 40 and redraw the binary tree.

```
```

```
    80
   / \
  65  85
 / \   \
35  75 ```
```

5. Delete nodes 80 and 58 in that order. Redraw the binary tree after each deletion.
   بعد حذف ٨٠:

```
    75
   / \
  65  85
 / \   \
35  58 ```
```

بعد حذف ٥٨:

```
    75
   / \
  65  85
 /     \
35      ```
```

4- Write the definition of the function, **nodeCount**, that returns the number of nodes in a binary tree.

```
5- // Binary Tree Node class
6- class Node {
7-     int data;
8-     Node left;
9-     Node right;
10-
11-    Node(int data) {
12-        this.data = data;
13-        this.left = null;
14-        this.right = null;
15-    }
16-}
17-
18-// Binary Tree class
19-class BinaryTree {
20-    Node root;
21-
22-    // Constructor
23-    BinaryTree(Node root) {
```

```
        this.root = root;
    }

    // Recursive function to count nodes in a binary tree
    public int nodeCount(Node node) {
        if (node == null) {
            return 0;
        } else {
            return 1 + nodeCount(node.left) + nodeCount(node.right);
        }
    }
// Wrapper function to call nodeCount with the root of the tree
    public int getNodeCount() {
        return nodeCount(root);
    }
}
```

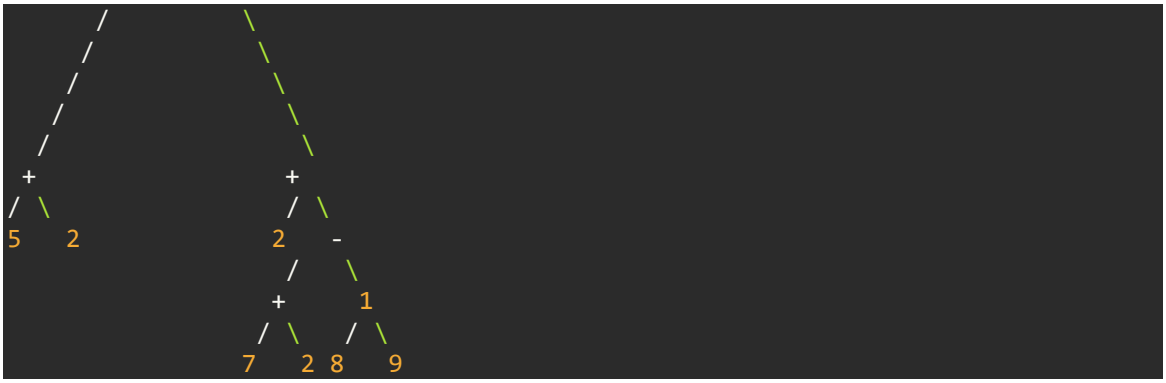
40- Write the definition of the function, **leavesCount**, that takes as a parameter a reference to the root node of a binary tree and returns the number of leaves in a binary tree.

```
class Node {
    int data;
    Node left;
    Node right;

    Node(int data) {
        this.data = data;
        this.left = null;
        this.right = null;
    }
}

// تعريف الدالة لحساب عدد الأوراق
public int leavesCount(Node root) {
    if (root == null) {
        return 0;
    } else if (root.left == null && root.right == null) {
        return 1;
    } else {
        return leavesCount(root.left) + leavesCount(root.right);
    }
}
```

63-

64- Draw the binary tree representation of the following arithmetic expression:
"(((5+2) ∗ (2−1))/((2+9)+((7−2)−1)) ∗8)".

```
*
          /   \
        /      \
       /        \
```



65- Insert, into an empty binary search tree, entries with keys 30, 40, 24, 58, 48, 26, 11, 13 (in this order). Draw the tree after each insertion.

بعد ادخال ٣٠:

بعد ادخال ٤٠:

بعد ادخال ٢٤:

بعد ادخال ٥٨:

بعد ادخال ٤٨:

بعد ادخال ٢٦:

```
   /
  11
```

بعد ادخال ١٣:

```
30
    /   \
   24    40
    \     \
    26    58
   /       \
  11        48
   \
    13
```

يمكن حل أسئلة من الكتاب ، بالإضافة الى المحاضرات

Good Luck