Binary Tree Assignment

Marks: 25

Modify the BTree class developed in class to implement the following methods. The methods should be as easy to use as possible, for example you should not need to pass in the root of the tree to use a display method.

1. **depth**

   Takes an integer and returns the depth where integer is found. Returns -1 if not found.

   The depth of a node in a binary tree is the length of the path from the root of the tree to that node. (The root has depth 0, its children have depth 1, its grandchildren have depth 2, and so on.)

   (2 marks)

2. **display**

   Prints all elements in the tree. This method must be overloaded to take either no parameters or a single integer. The integer must be defined as a constant in the BTree class. These constants must be IN, PRE and POST and control if the tree is to be displayed in-order, pre-order or post-order. If no parameter is specified then display the tree in-order. (2 marks)

3. **search**

   Write a method that takes an int value and searches for a node with a specific value and returns the **node (Bnode)** if found. (2 marks)

4. **sumLeaves**

   Returns the sum of the values of the leaf nodes in the tree. (2 marks)

5. **total**

   returns the sum of all the nodes in the tree (2 marks)

6. **height**

   Returns the height of the tree. (maximum depth) (3 marks)

7. **isIdentical(BTree otherTree)**

   Returns true if the tree is identical to another tree and false otherwise

   (4 marks)

8. **isBalanced**

   Returns true if the tree is balanced and false otherwise. (4 marks)

9. **add(BTree otherTree)**

   Overload add so that it takes a BTree. (Add copies of all of the nodes of the BTree parameter to the current tree.) (4 marks)