Loyola University Chicago
Department of Computer Science
COMP 272: Data Structures II (Spring 2024)

Assignment # 3

This is an individual assignment.
Deadline: **Wednesday, February 21st, 2024, 11:55PM.**
You need to submit your solutions in Repl.it.

AVL Trees (10 points)

Complete the *deleteElement()* method. This method has two parameters: integer value, Node node.

The purpose of the method is to delete an element within the tree.

This exercise is more difficult than the ones below, however this exercise is **more real life**, where you are to modify existing application by adding functionality. This is a running AVL Tree that had the method *deleteElement()* removed, yet I kept the method's prologue comment there on what it does. Those are your requirements on implementing the method.

I realize this one is more difficult than below, so kept the points to 10, but in reality, this is more real life and what is expected of an entry level developer.

Binary Search Tree (30 points)

Implement the following methods:

- **insertAll**

Complete the *insertAll()* method. This method should accept an array of integers and insert each of them into the tree. It should invoke a private method *insert()* which you need to code as well.

- **increaseAll**

Complete the *increaseAll()* method.

The purpose of this method is to traverse each node in the tree and increase the value of the node's data by the integer parameter passed to the method.

Tree Set (30 points)

Complete the *different()* method. This method should return a set containing elements that are different between the two input Tree Sets.

Recommend using the TreeSet methods *retainAll*(), *addAll*() and *removeAll*() to implement this *different()* method. If doing so, will only take a few lines. But in the end, just get working.

Tree Map (30 points)

Complete the *removeEven()* method. This method should remove entries from a given TreeMap where the key is an even number.

This method should remove the elements from the *TreeMap* passed as a parameter. Recommend Using a *TreeMap* iterator object and simply loop through the map removing even number keys.

Submission:

1) Before submission, make sure your code passes all the JUnit tests. Keep in mind, however, that passing the test cases does not guarantee that your code is correct or efficient. Your assignment will be graded considering test results, correctness, and efficiency.
2) The submission should be completed in Replit.
3) Include your name and class number as comments at the top of each submitted Java file.