CIS 044:   
Introduction to Data Structures Using Java

Lab 9

**Instructor**

Dr. Maher Mneimneh  
Email:mahernm@gmail.com

**Guidelines**

**Please follow the guidelines below:**

**Submit the homework and lab solutions in the drop box. For a programming question, submit a .java file (for source code) and a .txt file for program output. If the problem involves other questions, submit a separate .txt file to answer the question.**

**P1 (30 points)**

Suppose we want to create a method for the class **BinaryTree** (file **BinaryTree.java**) that counts the number of times an object occurs in the tree.

1. Write the method

**public int count1(T anObject)**

which calls the private recursive method

**private int count1(BinaryNode<T> rootNode, T anObject)**

to count the number of occurrences of **anObject**

1. Write the method

**public int count2(T anObject)**

that counts the number of occurrences of **anObject** and that uses one of the iterators of the binary tree.

Compare the efficiencies of the previous the two methods **count1** and **count2** using big O notation. Add your answer as a comment before the function definition

**P2 (30 points)**

Suppose we want to create a method for the class **BinaryTree** that decides whether two trees have the same structure. Two trees t1 and t2 have the same structure if:

* If one has a left child, then both have left children and the left children are isomorphic, AND
* if one has a right child, then both have right children and the right children are isomorphic

The header of the method is:

**public boolean isIsomorphic(BinaryTree<T> otherTree)**

Write this method, using a private recursive method of the same name.

**P3 (40 points)**

Design an algorithm that produces a binary expression tree from a given infix expression. You can assume that the infix expression is a string that has only the binary operators +, -, \*, / and one-letter operands. Implement the solution as a construction in ExpressionTree that takes a string argument:

**public ExpressionTree(String infix)**

that calls the private method

**private ExpressionTreeInterface formTree(String expr, int first, int last)**

to construct the tree**. formTree()** builds the tree recursively.