CS1083 Assignment # 11 - Fall 2024

Due: Wednesday, 4 December before 4:30 pm in the Desire2Learn dropbox. (See submission instructions below).

The purpose of this assignment is:

- Review Binary Search Trees

This assignment is to be done individually. If you have questions, direct them to a tutor/assistant during an extra help session. If your question is not answered during an extra help session, you may contact your course professor.

Customer Class

For this assignment, you re provided with the file Customer.java on D2L which represents a customer at a store. You will use the Customer class for this assignment.

BinarySearchTree and BSTNode

Create the class BinarySearchTree that contains the private sub-class BSTNode. For BSTNode, there should be 4 instance variables, a left and right BSTNode, a Customer, and a frequency (frequency should be initialized to 1).

BinarySearchTree will have a root BSTNode. The following methods must be implemented for BinarySearchTree:

- `public boolean insert(Customer c)`

  If the given Customer is not already stored in a node in the tree, this method will insert the Customer in the Binary Search Tree in the correct ordered position and return true. If the Customer is already in the list, increment the counter associated with the BSTNode storing the matching Customer and return false.

  Note: You may also want a helper recursive method:
  `private boolean insertRec(BSTNode curr, BSTNode toAdd)`

- `public int search(Customer c)`

  Searches the tree for a BSTNode that stores data that equals the Customer passed in as a parameter. If the data is found in the tree, return the frequency of that Customer. Otherwise, return a 0.

  Note: You may also want a helper recursive method:
  `private boolean searchRec(BSTNode curr, BSTNode toFind)`

- `public void printInOrder()`

  Prints the BST in order using Customer's toString method and the frequency.

  Note: You may also want a helper recursive method:
  `private void printInOrderRec(BSTNode curr)`

- `public void printPreOrder()`

  Prints the BST in pre-order using Customer's toString method and the frequency.

  Note: You may also want a helper recursive method:
  `private void printPreOrderRec(BSTNode curr)`

- `public void printPostOrder()`

  Prints the BST in post-order using Customer's toString method and the frequency.

  Note: You may also want a helper recursive method:
  `private void printPostOrderRec(BSTNode curr)`

BSTDriver Class

Create a basic driver program to create various Customer objects and add them to a BinarySearchTree. Should try adding duplicates. Print the BST with all three methods to ensure they work. Finally, test the `search` method.

Your electronic submission (submitted via Desire2Learn) will consist of two files. Name your files YourName-fileName.extension, e.g. JohnSmith-as11.zip, JohnSmith-as11.pdf:

1. A single pdf file containing a listing of the code for your program.
2. A zip file containing all your Java classes.