Assignment 4
Tree and Heap

Release Date	Due Date
11/3/2017	**11/14/2017**

Objectives

- To practice building and traversing trees and heaps.
- To develop high-performance solutions.

Problem Specification

Yu is a teaching assistant in WMU, and his supervisor asks him to write a program to manage the students in CS3310. Please help him develop such a program with appropriate data structures. Your tasks include the following.

1) Read the data from *NameList.txt* file. Each line consists of a name in the format: LastName FirstName.
2) Build a min-heap using a size balanced binary trees that is ordered by LastName. Note that size-balanced binary tree uses explicit representation with each node having at least the following data members: parent, leftchild, rightchild, sizeOfSubtree, and data (data may be composed of multiple components/parts).
3) Build a max-heap using an array-based min-height binary tree (ABT) that is ordered by firstName.
4) Build a Binary Search Tree (BST) using an explicit representation that is ordered by LastName.
5) Please use pre-order, in-order, and post-order traversal to print your min-heap, max-heap and BST.
6) Locate (i.e., find) your name in the three data structures, and return the location (which level, which node position counted from left to right). For the max-heap and min-heap, you must first use breadth-first search (BFS) and then use depth-first search (DFS) (that is level-by-level search).
7) Record the time used to locate your name in the three data structures.

Design Requirements

Code Documentation

For this assignment, you must include documentation for your code as generated by JavaDoc. You should have JavaDoc comments for every class, constructor, and method. By default, JavaDoc should output html documentation to a subfolder within your project (/dist/javadoc). Make sure this folder is included when you zip your files for submission. You do not need to submit a hard copy of this documentation.

Hint: http://stackoverflow.com/questions/4468669/how-to-generate-javadoc-html-in-eclipse

Coding Standards

You must adhere to all conventions in the CS 3310 Java coding standard. This includes the use of white spaces for readability and the use of comments to explain the meaning of various methods and attributes. Be sure to follow the conventions for naming classes, variables, method parameters and methods.

Testing

Make sure you test your application with several different values, to make sure it works. Testing: test your program with several different text files containing different datasets, design your test cases so that you cover most (if not all) cases of user-input (this implies your application should be robust and fail-safe as much reasonably as possible).

Assignment Submission

- Generate a .zip file that contains all your files, including:
  - Source code files
  - Including any input or output files
  - Documentation of your code – e.g. using Javadoc if using Java
  - A brief report (in a pdf file) on your observations of comparing theoretical vs empirically observed time complexities. Note this report will include (a) a brief description of problem statement(s), (b) algorithms descriptions (if these are standard, commonly known algorithms, then just mention their names along with customization to your specific solution(s), otherwise give the pseudo-code of your algorithms, (c) theoretically derived complexities of the algorithms used in your code, (d) table(s) of the observed time complexities, and (e) plots comparing theoretical vs. empirical along with your observations (e.g. do theoretical agree with your implementation, why? Why not?).
- Don’t forget to follow the naming convention specified for submitting assignments