Fall '14 CIS 212 Assignment 5 – 110/100 points possible – Due Wednesday, 11-5, 11:59 PM

The goal of this assignment is provide experience reading data from a file and sorting data using Selection Sort and Merge Sort. This project will involve building a simple UI to evaluate the efficiency of the two sorting algorithms.

When debugging your application, I highly recommend that you start by sorting a small subset of the total entries (e.g., only the first 10 entries). This will allow you to follow a complete run of your sorting algorithms, providing opportunities to spot inconsistencies with respect to expected behavior.

1. [20] Write a Java application which first reads data from the phonebook file here: http://www.cs.uoregon.edu/Classes/14F/cis212/assignments/phonebook.txt

into an ArrayList of phonebook entries. We'll need to differentiate between the name and phone number when sorting and retrieving data, so each entry should consist of separate fields for the name and phone number (e.g., a class with two private String variables).

2. [20] Implement a method which takes an ArrayList of phonebook entries as an argument and returns a sorted copy of the list using Selection Sort to sort alphabetically by phonebook name. Your implementation should not modify the input list. You should implement your own sorting code here, and not simply use built-in Java calls (e.g., in the Arrays and Collections classes). Cite any sources used in the comments of your code.

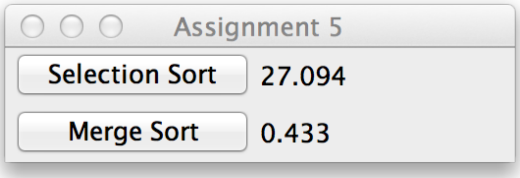
3. [20] Implement a method which takes an ArrayList of phonebook entries as an argument and returns a sorted copy of the list using Merge Sort to sort alphabetically by phonebook name. Your implementation should not modify the input list. Again, you should implement your own sorting code here, and not simply use built-in Java calls (e.g., in the Arrays and Collections classes. Cite any sources used in the comments of your code.

4. [20] Implement a method which takes an ArrayList of phonebook entries as an argument and returns true if the input list is sorted alphabetically by phonebook name (false otherwise). Use this method to test your sorting implementations.

5. [20] Implement a GUI with at least two buttons:

1. A button to call your Selection Sort method from part 2 above and report the elapsed sorting time in the GUI. Clicking this button should report "Error" if the resulting list is not sorted according to your method from part 4 above.
2. A button to call your Merge Sort method from part 3 above and report the elapsed sorting time in the GUI. Clicking this button should report "Error" if the resulting list is not sorted according to your method from part 4 above.

Your window should look something like:



6. [+10] (Extra credit) You’ll notice that the GUI is unresponsive while sorting, which is because each sorting processes is started as a result of an event and is therefore executed on the UI thread. To avoid this issue, create and execute a new java.lang.Thread for each sorting process.

Zip the Assignment5 folder in your Eclipse workspace directory and upload the .zip file to Blackboard (see Assignment 5 assignment in the Course Documents area).