CS 2401 Assignment #7

Due Date: Friday November 03, 2017 11:59PM (See the syllabus for late policy).

Objective: The goal of this assignment is to practice sorting algorithms.

Assignment: In this assignment, you will be implementing versions of two sorting algorithms. You will use them to sort an array of strings in lexicographic order. All other requirements are listed below.

- You should write a method in a class named `Runner1` that will generate an array of random strings for testing. The method should take in a number representing the size of the list, and return an array of that many Strings. Each string should have a randomly generated length between 3 and 10 (that is, $3 \leq length \leq 10$).
- Implement another method in `Runner1` to print out your array of strings. You should print a line for each string.
- As another preliminary step, implement a method `CopyArray` in `Runner1` that creates a new array of strings with identical values of a given one.
- Now, write two methods inside `Runner1`, one named `BubbleSort` and the other named `SelectionSort`. Use parameters as you think appropriate. `BubbleSort` must implement the bubble sort algorithm we discussed in the class and `SelectionSort` must implement the selection sort algorithm we discussed in class. Note that both `BubbleSort` and `SelectionSort` must sort the strings given in the input array in lexiographic order.
- Demonstrate that your code works by generating a random list of 20 strings (in an array), printing out the list before you sort it, and then printing out the list again after it is sorted using `BubbleSort`. For an identical list of 20 strings (copied right after the generation of the array that was given to `BubbleSort` as an input), show the list before and after the use of the `SelectionSort` method.
- Write a class named `Runner2` that will test the speed of your `BubbleSort` and `SelectionSort` methods by recording the time needed to sort an array of different number of Strings (For example, 2000, 3000, 4000, 5000, 6000, and 7000 Strings). Using this data, draw two lines on the same line-plot using MS Excel, one line for Bubble Sort and one for Selection Sort, where x-axis is the number of Strings in the array and y-axis is the runtime. On the MS Word document, (1) copy the plot and paste it, (2) report the configuration (processor speed and model, memory, and operating system) of the computer where you executed `Runner2`, (3) write what are the time complexities of Bubble sort and Selection sort algorithms, (4) write no more than four sentences on if (and how) your plot demonstrates the expected time complexities of Bubble sort and Selection sort algorithms. Submit the MS Word file with your source codes. Convert/Export your MS Word file to a PDF file and submit it with all other files.

Deliverables: `Runner1.java`, `Runner2.java`, `MyPlot.docx`, and `MyPlot.pdf`. You must use Blackboard to submit. Talk to your TA for further instructions.