CS310 Data Structures Spring 2020

Programming Assignment 2

Sorting

For this project, you will implement three sorting algorithms, and then test their performance. This project will consist of not just coding, but also testing your code using large data sets.
Coding Sorting Algorithms
For Part 1 of the assignment, you will need to write 3 sorting algorithms:

- Insertion Sort
- Quick sort
- Merge Sort

All three sorts must work on a LinkedList (Use LinkedList from Java Util) of type Integer.

Class Description

- Sorting - Interface - Contains definitions of methods for the sorting algorithm
- SortingTest- Class - Implements the Sorting interface and also contains the driver class.

Constraints

- Your SortingTest class has a constructor that takes no parameters
- Your Sort class implements the interface as is, without changes.
- You are allowed to write as many extra (private) helper functions as you would like.

Efficiency Testing

After you have coded your algorithms, you need to test them, to see how long each sorting algorithm takes to run.
You should test each algorithm on both random and sorted lists (ascending & reverse sorted) of sizes 10, 50, 100, 150, 300, 500, 1000, 1500
You can get a random list using the java class Random, located in java.util.Random To test the speed of your sorting algorithms, you should use the System.currentTimeMillis() method, which returns a long that contains the current time (in milliseconds).
Call System.currentTimeMillis() before and after the algorithm runs, and subtract the two times. Unfortunately, using currentTimeMillis before and after a function only gives an accurate time estimate if the function takes a long time to run (that is, at least a couple of seconds).
You should subtract the setup time from your algorithm running time, to get more accurate results. i.e. only calculate the time taken for the main loop which runs the sort.
Your main program may be interactive (i.e. give options to the user to run each individual sort) or automated where all results are displayed without user interaction.

Program Flow / Expected output:

When you run the SortingTest program the following must happen:

Run each sorting algorithms with the following parameters:

1. With a Sorted LinkedList, generate the time for the following number of outputs:

   10, 50, 100, 150, 300, 500, 1000, 1500

2. With a Reverse Sorted LinkedList, generate the time for the following number of outputs:

   10, 50, 100, 150, 300, 500, 1000, 1500

3. With an unsorted LinkedList, generate the time for the following number of outputs: 1000, 10, 50, 100, 150, 300, 500, 1000, 1500

ReadMe

Showcase the results in a tabular form like below for all three algorithms generated by your program

Sort & Input	Sorted Linked List	Unsorted Linked List	Reverse sorted Linked List
QuickSort			
10			
50			
100			
150			
300			
500			
1000			
1500			

Mention the worst and best cases of each of the sorting algorithms from the textbook and compare them with your results, there by proving the Big-O of each of the algorithms for best, average and worst cases.

Files to be submitted:

Submit a ZIP file (ZIP file must have your First and Last Name) containing the following: 1.

1. Folder with all your source files. (NO JAR FILE OR BATCH FILE IS NECESSARY)
2. ReadMe.pdf - with all the questions above answered.

A project folder is provided for you. Use the Interface as is without changes.

DUE: 18th April 2020