ENSF 594 – Principles of Software Development II

Summer 2021

Lab Assignment # 2

Comparison of Sorting Algorithm

Total Marks: 25

Due Date: Monday, July 19, by 11:59 PM using D2L

Computers are frequently used to sort lists of items, especially when these lists are long. Many sorting techniques are available, some more efficient than others. In this assignment, you will implement four different sorting algorithms in a Java language program and compare their performance. The sorting algorithms you will implement are:

1. Bubble Sort
2. Insertion Sort
3. Merge Sort
4. Quick sort

You can base your own code for these algorithms on the code discussed in lectures, or on code found from other sources. Be sure to cite the source for any code you borrow or adapt, putting the citation in comments in your program

Your program will create an array of integers to be sorted. The user will be able to specify whether the array is filled with integers in random order, ascending order, or descending order. The length of the array is arbitrary and will be specified at run time. The array may contain duplicate numbers. Your program will take this array and sort it into ascending order, outputting the sorted list to a text file, one item per line. The sorting algorithm to use and the name of the output file will also be specified at run time from the user.

Your program will time how long it takes to sort the array. Be sure to time only the sorting function itself, and not the time taken to fill the array with numbers or do input or output. Your program will print the time in seconds to the screen (standard output). You will use this data to help compare the efficiency of the four algorithms.

Write a program in the Java programming language to implement the above requirements. Your program will take the following input:

order is the order of the integers in the input array (use one of the following strings: ascending, descending, random), ***size*** is the number of items in the integer array to be sorted, ***algorithm*** specifies the sorting technique to use (use one of the following strings: bubble, insertion, merge, quick), and ***outputfile*** is the name of the output file where the sorted list will be written to.

Experiment and Data Collection:

You will use your program to perform a series of experiments that help you to compare the efficiency of the four sorting algorithms in the best case (input array already sorted into ascending order), worst case (input array in descending order), and average case (input array in random order). At the very least, use the following sizes for the array length: 10, 100, 1000, 10,000, 100,000, and 1,000,000. Determine the timings for all four algorithms using these inputs.

Complexity Analysis

Using the techniques shown in class, do a complexity analysis of each of the four algorithms by following the steps.

Report

Create a formal PDF written report that describes at least the following:

1. The data collected (use tables and graphs to help illustrate this).
2. Data analysis. What does the data tell us about the algorithms?
3. The complexity analysis
4. Interpretation. What does the empirical data and complexity analysis tell us about the algorithms? How do the algorithms compare with each other? Does the order of input matter? How do algorithms within the same big-O classification compare to each other?
5. Conclusions. What algorithms are appropriate for practical use when sorting either small or large amounts of data? Are there any limitations with any of the algorithms? How does your analysis support these conclusions?

You can add more to this, if necessary. Be sure the report is complete, organized, and well formatted.

Notes for Submission:

1. Your formal PDF written report.
2. Your source code files. Your TA will run your program to verify that it works correctly. Make sure your Java program compiles and runs without issues.