UNIVERSITY OF NOTH TEXAS
COLLEGE OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CSCE 4110 SPRING 2023– ALGORITHMS

Assignment 1 Sorting

Due Date: Friday February 17, 2023, at 5:00 PM

1   Goals

- Write Java code to sort a sequence of elements into ascending order by using the following sorting algorithms: Bubble, Selection, Insertion, Shell, Merge and Quick Sorts
- Write Java code to build graphical user interface GUI application.
- Assess the efficiency of various sort algorithms.

2   Description and Requirements

Graphical demonstrations of various sorting algorithms are instructive, as they provide insight into how an algorithm behaves. Consider a collection of vertical lines of varying lengths, such as the ones in Figure 1a. Create a sorting demonstration that sorts the lines by length, as shown in Figure 1b.

You should draw the configuration of lines after every swap or move that a given sorting algorithm makes. If you delay execution very briefly after each redraw, the result will be an animation of the sort.
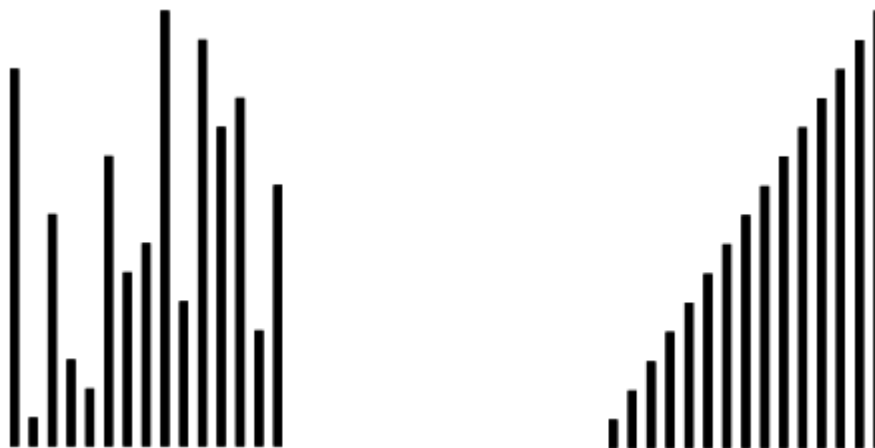

Figure 1: An animated sorting demonstration that sorts vertical lines
(a) before its execution (b) after its execution.

1. You need to read and exercise some of the examples provided in Unit-2 in the course lecture notes before starting to implement the requirements of this assignment.

2. You begin by drawing 256 lines, each one pixel wide but of different lengths and perhaps different colors arranged from shortest to longest so that they appear as a triangle.

3. The user then should exercise an option (one button click) to scramble the lines. At anyone click you should generate a new random scrambled lines, and then disable the action of this button until the all sorting methods are applied.

4. At a user signal, (button click, or radio button selection) your sorting algorithm should sort the lines.

5. When one sort method is finished, the user should have an option to reset the sorted lines back as they were before applying this sorting method.

6. When all sorting methods have finished, the scrambling button can then be enabled and used to generate another random scrambled lines.

7. This means that, each sort should start with the same scrambled lines so the user can compare methods.

8. Consult your TA for any point, specifically to your code as Java point of view.

3   Hand In

You need to use IntelliJ IDEA to create a Java Project and name it as "code_yourEUID_assignment#" (e.g., if EUID is aho0017 and you are submitting assignment 1 hence the IntelliJ project name will be "code_aho0017_assignment1"). Then create a Java package in your IntelliJ Java project and name it the same name as the project name. Under this package you will create the Java classes and interfaces. You should write all your code and documentation under this package. You need to ZIP the IntelliJ project (folder) and submit it electronically through Canvas.

4   Grading

Description	Points
Code documentation	10
GUI + Scrambling method	20
Bubble Sort	10
Selection Sort	10
Insertion Sort	10
Shell Sort	10
Merge Sort Iterative (Given)	-
Merge Sort Recursive	15
Quick Sort Recursive	15
Total	100

5 Assignment Sample Snapshots

Please note the diferences between each of the following snapshots and implement similarly. These snapshots are taken by the order of the program execution.