

# Data Structures and Algorithms

## Project 1

### Sorting Algorithms Empirical Time Efficiency Comparison

(100 points total) 20% of final grade

The purpose of this project is for you to realize how different algorithms solving the same problem produce different computational results, and to experiment with empirical analysis of algorithms.

The task is to conduct benchmarking to compare six sorting algorithms: bubble sort, selection sort, insertion sort, merge sort, quicksort, and shellsort. Each of the algorithms need to be run with random data sets of 10,000 50,000 150,000 200,000 275,000 items, 5 times each, averaged. To ensure proper distribution, the random data sets are to be selected from a range of 3,000,000 items. These data sets are to be placed in different files, based on the number of items in each file. All algorithms are to be tested with the same data sets.

Grading rubric

<i>Task</i>	<i>Points</i>
Bubble Sort implementation	5
Selection Sort implementation	5
Insertion Sort implementation	5
Merge Sort implementation	5
Quicksort implementation	5
Shellsort implementation	5
Grouping all Sorting algorithms in one implementation	5
User interface for algorithm and input file selection	5
Data sets generation	10
Detailed Report	50
Extend Report to include specific observations	15 (extra credit)

Grading details:

- (30 points) each Sorting algorithm implemented and documented properly; implementations that do not compile or do not work properly are worth 0 points
- (5 points) grouping all Sorting algorithms into one documented program
- (5 points) adding user interface to the program (could be text-based)
- (10 points) generation of data sets as detailed above
- (50 points) generating a quality Report which explains the findings from the experiments including graphs – you will be looking at comparing the time spent by

each algorithm on each data set and reporting why you think a specific algorithm works faster or slower; Report with no graphs is worth maximum of 10 points

- (5 points extra credit) expand the Report to include the following findings: which algorithm performs five times as fast as the slowest algorithm and for what data set size? how about twenty times, a hundred times? If the data does not allow you to provide an exact answer to any of these questions, provide an approximate answer based on the growth rate shown on the graphs
- (5 points extra credit) generate data sets which contain sorted increasing numbers and expand the Report to describe your findings
- (5 points extra credit) generate data sets which contain sorted decreasing numbers and expand the Report to describe your findings

#### Recommended Development Schedule

Week 1 (August 28 – September 3)	Individual Sorting Algorithms implementation
Week 2 (September 4 – September 10)	Algorithms revision and full program implementation
Week 3 (September 11 – September 17)	Data sets generation and data collection
Week 4 (September 18 – September 24)	Report
September 25 – September 28	Final revisions

Very Important: Experiments must be performed on the same computer to produce valid comparisons. Also, time only the execution of Sorting algorithms and not file reads.

#### Citations

If you use pieces of code that have been taken from published literature, then you must cite these sources in your Report. Otherwise, it is plagiarism. If you take ideas from published literature and improve upon them, cite the sources and explain your adaptation of those ideas to this project.

#### Introduction and Conclusion

Every technical report must have an introduction that briefly describes what the project is all about. (Please do not reproduce the project description here). Mention highlights of your program, if any, like special user-friendly features, or efficiency properties. The conclusion should contain any thoughts you may have about your current findings, and suggestions for possible future enhancements and modifications.

#### **What and How to Submit**

You need to submit all source files containing the implementation of your project, and the Report. The source files must be grouped in a zip file, and submitted in Canvas on the due date. The technical report must be in Microsoft Word format, and submitted in Canvas on the due date. Also submit a video demo of your project (if applicable).