Programming report exercise

Amanda Enhörning s1128126 Jessica Borg s1129470

7 september 2023

1 Introduction

Sketch the context of the work so that the reader understand what they are reading. In this case: write that you will empirically compare two sorting algorithms on a set of inputs to get an idea on the difference between worst-case and best-case running times for both algorithms. This is quite similar as the Introduction in the example report.

2 Specification

Specify the task at hand, with the assignment text as your guideline. Note that the algorithms are already given in this example exercise, but normally you would describe what the software should do (in this case: read in a comma-separated values (.csv) file with arrays, one per row of the file, sort the array using Selection Sort and Insertion Sort, count the array assignments and comparisons, and return the sorted array as well as the measures). This specification is in the first part of the Methods in the example report.

3 Design

Specify here your software design. Again, the algorithm is already given, but here you would specify the different classes and data structures you use, packages used for I/O and visuali- sation etc., and the pseudo code for existing algorithms that you will implement. In this case you would specify that you separate the sorting from the I/O (reading in the .csv file, writing down the results), you would give the pseudo-code for the sorting algorithms, etc. The middle part of the Methods in the example report contains this information.

4 Implementation

Give (only) the crucial part of the implementation here, preferably with line numbers, and explain what it does without literally repeating the code. Highlight important aspects and don't repeat trivial translations from the pseudo-code. This is in the last part of the Methods in the example report.

5 Testing

Explain how you will test and experiment on the implementations in order to get a good idea of 1) their correct working and 2) the empirical results you are interested in. In this case you would describe the contents of the .csv file (extremes, such as already sorted arrays, arrays sorted in opposite order, uniform arrays; as well as several randomly sorted arrays) and how you would interpret the results. Oftentimes the assignment will ask you for particular empirical questions that you can summarize here.

6 Results

Write up the results in a clear and easy to follow way. Use a table and if meaningful, a graph, to summarize results. Give the raw facts here, not the interpretation of the facts. For example, a table like this could be useful:

	Insertion		Selection	
	Assignments	Comparisons	Assignments	Comparisons
Already sorted	...	...	...	...
...	...	...	...	...

7 Discussion

In the discussion you interpret the results. What do you see and what does it mean (spoiler: Selection Sort has the same number of comparisons and assignments for every n-sized array, independent of the order, whereas Insertion Sort is sensitive to the order). The example report could still be improved somewhat by more clearly separating raw results (running times) and their interpretation.

8 Conclusion and Reflection

Conclude the report. Summarize the main findings, mirroring the introduction (in this case: Selection Sort has the same worst-case and best-case run-time complexity, whereas Insertion Sort is linear time in the best case). Reflect on the assignment: what obstacles did you encounter? Any interesting observations? What could still be improved? From this year on it is compulsory to give an overview of the work division. Any specific information can go here (e.g., building on existing partial implementation of one of the team members, someone dropping out or joining late, etc.). As a general rule: all team members should contribute to both coding and writing the report!