

EECS 2030: Lab 7 (the final lab in the course)

(1.5 % of the final grade, may be done in groups of up to three students)

Motivation

The purpose of this lab is to practice analyzing the running time of the algorithms.

Part 1: Getting Started

Download a zip file containing the Lab 7 Eclipse project.

Import the project into Eclipse by doing the following:

1. Under the **File** menu choose **Import...**
2. Under **General** choose **Existing Projects into Workspace** and press **Next**
3. Click the **Select archive file** radio button, and click the **Browse...** button.
4. In the file browser that appears, navigate to your download directory (exactly where this is depends on what computer you working on; on the lab computers the file will probably appear in your home directory)
5. Select the file **lab7.zip** and click **OK**
6. Click **Finish**.

Part 2: Tasks

The following three algorithms are to be analyzed:

mysterySort1

mysterySort2

mysterySort3

The `sortExperiment` class is provided to allow some preliminary observation of the sorting methods' behaviour. Feel free to modify it any way you desire to help you complete the assignment: you only need to submit a written report at the end. For reference, the default list sorting algorithm used in Java is an $O(n \log n)$ algorithm based on mergesort (*TimSort*).

For each algorithm, try to come up with an analytical solution for their running time complexity whenever possible. Use n , the size of the input list, as the problem size. Basing *all* your solutions purely on empirical analysis (i.e., merely observing how the running time changes with the growth of the list) will result in a lower grade.

The report should be typed: no handwritten and scanned text is accepted. If you wish, you may insert illustrations into your document. Please use proper English (e.g., no shortcuts like “WRT”, “2B”, “4” “u”, or “RTFM”), appropriate punctuation and formatting, and proofread your writing before submitting the report.

Helpful links

Sorting algorithms, https://en.wikipedia.org/wiki/Sorting_algorithm

Table of common time complexities,

https://en.wikipedia.org/wiki/Time_complexity#Table_of_common_time_complexities

If you have questions, or think that you found an error, don't hesitate to post your questions on the course forum on Moodle, or contact the instructor directly (andriyp@cse.yorku.ca).

Submission

Submit your report as a single PDF or DOCX file electronically via Moodle. Again, the report should be typed: no handwritten and scanned text is accepted.

If working in a group, make only one submission and include a `group.txt` file containing the names and the student numbers of the group members. The deadline is firm. Contact the instructor *in advance* if you cannot meet the deadline explaining your circumstances.

Grading

The assignment will be graded using *the Common Grading Scheme for Undergraduate Faculties*¹. The report will be judged both on the correctness of the solution and on the clarity of the supporting explanation.

Academic Honesty

Direct collaboration (e.g., sharing your work results across groups) is not allowed (plagiarism detection software may be employed). However, you're allowed to discuss the assignment requirements, approaches you take, etc. Also, make sure to cite any sources you use (online sources – including web sites; old solutions; books, etc.)

¹ <https://secretariat-policies.info.yorku.ca/policies/common-grading-scheme-for-undergraduate-faculties/>