

Practice Assignment 4 - Sorting (II)

The goal of this assignment is to practice the implementation of the sorting algorithms we discussed and to determine the efficiency as measured by (unreliable) wall clock time.

Background

We discussed two algorithms for quasilinear sorting: merge sort, quicksort. These algorithms have the same average running time — $O(n \lg n)$. This assignment will have you implement these algorithms as functions in different classes and test the timing of the implementations against each other. Insertion sort, as implemented in the previous practice assignment, is also used as a comparison.

Requirements (Process)

Requirement 1: Get the files you need. You will copy them to your own GitHub repository using the following procedure:

1. Log into GitHub. If you do not have a GitHub account, create one via github.com > Sign Up.
2. Point your browser to the URL <https://classroom.github.com/a/fsbJo6hz>.
3. If necessary, authorize GitHub Classroom by selecting the “Authorize github” button.
4. If available, select your name from the list of names available. This will link your GitHub ID.
5. Accept the assignment by selecting the appropriate button.

If successful, your repository should contain three (3) Java files:

- Practice04Test.java — the class containing the main function
- SortingFactory.java — the factory for the class Practice04Search
- SortingAlgorithm.java — the interface for the sorting classes

Requirement 2: Add to the code in order to make it run.. In order to get the code working, you will minimally need to implement three missing classes for sorting. In alphabetical order, these files are: `InsertionSort.java` (which you may re-use from the previous practice assignment), `MergeSort.java` and `QuickSort.java`. Each of the classes must implement the interface defined in `SortingAlgorithm.java`, specifically its function: `sort`.

Requirement 3: Produce a graph of running times. In order to do this, you will need to execute the main file, which will run your implementations 10 values in the Practice04Test class. You may use any software to produce this graph, including Google Sheets. An example is shown in Figure 1.

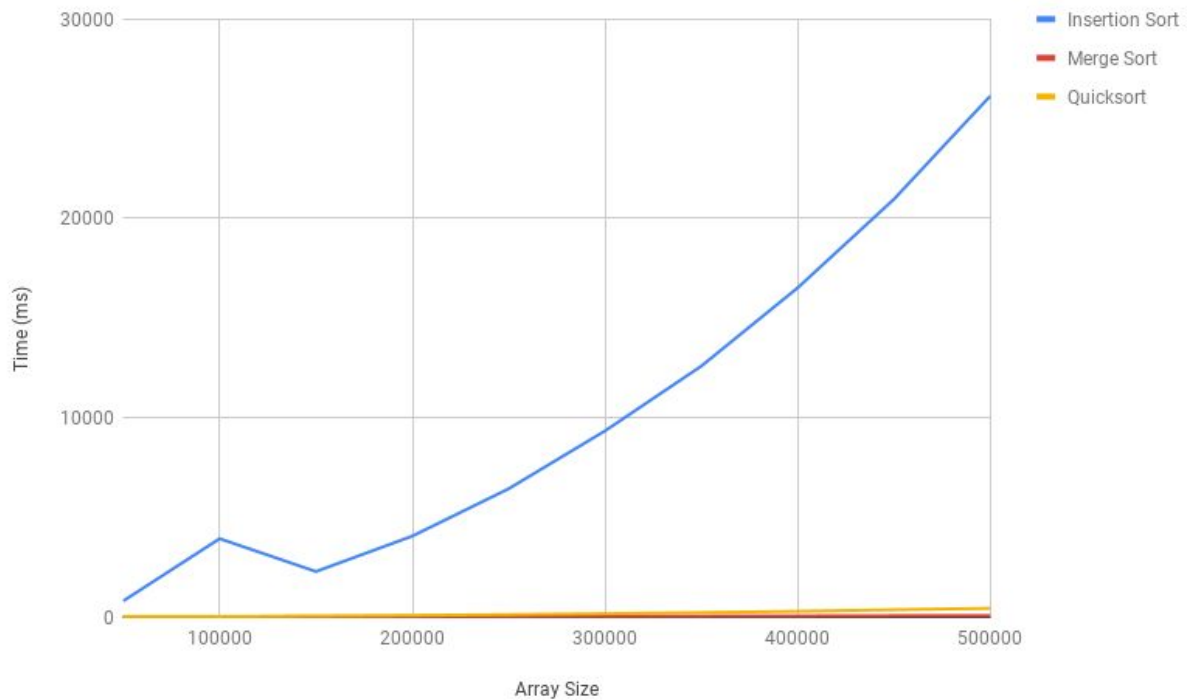


Figure 1: Graph of run timing

Submission

You are required to submit two items for this assignment:

- 1) Code to complete the functions in the implementation consisting of three sorting algorithms: Insertion Sort, Merge Sort and Quicksort.
- 2) A graph showing the (wall clock) running time of the algorithms, as is shown in Figure 1.

Use GitHub to check in the code to complete the implementation. On Canvas, submit the URL for your GitHub repository. Use a Google Sheets document or similar to record your (wall clock) running time and to graph the running times. You may link to the Google Sheets document from your GitHub repository in a README.md file.

Grading

Grades will be determined as follows:

- 75% = Correct implementation of each sorting algorithm (25% each)
- 15% = Graph with all (30) data points (0.5% each data point)
- 10% = Style, including consistent indentation, variable naming, etc.