

GTU Department of Computer Engineering
CSE 222/505 - Spring 2023

Homework 7

Due date: May 20, 2023 – 23:59

In this homework you are expected to extend your HW6 solution with various sorting algorithms.

Part 0:

The solution of HW6 will be shared with all students after its deadline. You can use this solution or your own work for HW6.

Part 1:

For a number of inputs use the following sorting algorithms to sort myMap structure.

- a) Merge Sort.
- b) (15 pts) Selection Sort
- c) (15 pts) Insertion Sort
- d) (15 pts) Bubble Sort
- e) (15 pts) Quick Sort

You are expected to implement all these algorithms (other than Merge Sort, which was implemented in HW6) by yourself. You should create new classes for the new sorting algorithms. The allowed libraries and variables you can use are the same as in HW6.

Part 2:

You should write a report containing the following information:

- a) (10 pts) Best, average, and worst-case time complexities analysis of each sorting algorithm. Make sure you explain your answer.
- b) (10 pts) Running time of each sorting algorithm for each input. You can add a method to each sorting algorithm's class to measure the time.
- c) (10 pts) Comparison of the sorting algorithms (by using the information from Part2-a and Part2-b). Which algorithm is faster in which case?
- d) (10 pts) In HW6, it was highlighted that the ordering of the letters with same count value should be the same as their addition order to myMap object. However, for these 4 sorting algorithms, the case might not be the same. You are expected to analyze which algorithms keep the input ordering and which don't, along with the code snippet that causes/ensures this.

Inputs

You should run all sorting algorithms for 3 different inputs (which can be declared within the code).

- 1) An input which will trigger the best-case scenario.
- 2) An input which will trigger the average-case scenario.
- 3) An input which will trigger the worst-case scenario.

Important

- Other than the ordering of same count values (as stated in Part2-d) and the content of report, the constraints are the same with HW6.
- The input strings don't have to be meaningful. For example, "xxxxx yyyy" is accepted as long as it meets one of the cases specified above.
- There isn't a single average-case scenario. Your input doesn't have to be 'perfectly' average. Something in between best and worst is enough. Similarly, you can write infinite best/worst case scenarios. But you are expected to write a single input for each case.
- Make sure your inputs are proper in order to get full credit.

Grading Details

No OOP Design	-100 points
No error handling	-50 points
No report (details are given above)	-40 points
No javadoc documentation	-50 points
Cheating	-200 points
Using any library other than the ones stated above	-50 points

Grading of Part 1: As in HW6 you may get partial credit for each sorting algorithm with regard to the correctness of your solution.

Grading of Part 2: The grade of each sub-problem will be equally distributed to each sorting algorithm. For example, 2 points for the asymptotic analysis of Merge Sort.