

CS2401 – Week 7

With this lab assignment, you are going to get a chance to practice on sorting algorithms and time complexity. This lab assignment should be reasonably straightforward. We need you to read it very carefully before you start doing anything. You should be able to rephrase the instructions and split the lab into smaller units before you start. Let's get started!

What is the goal of this lab?

We expect that, by the time you complete this lab you will:

- Be more confident about designing and testing sorting algorithms; and
- Better understand the concept of time complexity.

What problem will you be addressing in this lab? In this lab, you are going to write code to solve the following problem:

Given an array of integers,
you have to modify this array (not create another one) that is **sort of sorted**.

What do we mean by “**sort of sorted**”?

In this lab, what we mean by “**sort of sorted**” array is an array where:

3 rd largest	4 th largest	7 th largest	8 th largest	Etc.	6 th largest	5 th largest	2 nd largest	1 st largest
----------------------------	----------------------------	----------------------------	----------------------------	-------------	-----	----------------------------	----------------------------	----------------------------	----------------------------

For instance, given the following array of integers, called **numbers**:

2	7	1	3	0	9	6	5
---	---	---	---	---	---	---	---

Your algorithm will transform **numbers** into the following array:

6	5	1	0	2	3	7	9
---	---	---	---	---	---	---	---

What do you have to do?

1/ Write a method to transform the array of integers passed as a parameter into a sort of sorted array.

Your method called sortOfSort should take an array of integers (positive and/or negative; duplicates are also allowed) and transform the input array so that it is sort of sorted after running your method.

Note: your method sortOfSort should be a void method inside a file called

SortOfSort.java

2/ Test this method on at least 5 test cases using JUnit test cases. For this, you will create a file called **SortOfSortTester.java**. You are expected to describe each of the test cases in java comments right before each test case's code.

3/ Analyze the performance of your method sortOfSort. You are expected to clearly explain / justify your answer to this question. Your answer should address the best / worst / and average cases for the performance of your method. You will describe in a word document called **YourLastName-YourFirstName-Week7.docx** the performance of your sortOfSort method in terms of n , the length of the input array.

What should you turn in?

You should submit 1/ SortOfSort.java, 2/ SortOfSortTester.java, 3/ a word document called YourLastName-YourFirstName-Week7.docx in which you describe and clearly justify the performance of your method sortOfSort in terms of n , the length of the input array.

How should you submit your work?

You should submit on Git. However, if you are not yet fully familiar with Git, you can submit using the link provided by your TA (for a penalty of only 3 pts).

Failing to follow submission instructions and guidelines given by your respective TA will result in up to 15 points off your overall grade in this lab. So please pay attention.

Additionally, your word and java files are expected to be neat and clear (organization, grammar, and spelling for the word file / indentation and clear, meaningful variable naming for the java file). Failing to do so will result in up to 15 points off. On the other hand, extra neat and clear work will be rewarded by up to 10 extra points.

By when should you submit your work?

Due date: Friday March 6 at 5 pm

Lateness rule: -10 pts for 1 day of lateness / - 20 pts for 2 days of lateness / 0 after that, but you still have to turn in your work.

Grading:

40 pts Method sortOfSort

40 pts JUnit tests for method sortOfSort in SortOfSortTester.java

20 pts Analysis of performance