Kristianstad University
SE-291 88 Kristianstad
Sweden
+46 44 250 30 00
www.hkr.se

DA256B, DT256A:
Algorithms and Data Structures

Kamilla Klonowska

Seminar 1

Goals

The goal of this seminar is to prepare the students to fulfill the learning outcomes

Skills and abilities

- be able to apply the algorithm theory and data structures in practice (3)
- be able to describe and discuss its own expertise with various student groups (5)
- be able to work independently and in groups (6)

The students train: *Personal and professional skills and attributes*: Analytical reasoning and problem solving, as well as *Interpersonal skills*: teamwork and communication. (CDIO goals).

Before the Seminar –

The seminar is mandatory. A student that did not upload the tasks before the seminar is not allow to participate in the seminar.

All four tasks must be solved and they should be solved individually.

It is not sufficient to only solve the tasks, the student must also be able to explain the source code, explain how each algorithm and method works and discuss the results otherwise the tasks will not be approved. Note that hints of how some of the different algorithms work or how to implement them can be found in the course literature [1]. Make sure that for all other solutions that you find on the Internet and used as inspiration you have to refer to the source. Make sure to understand the algorithms before implementing them. Remember having your laptop with you in the seminar!!

Tasks

Implement a

Task 1: **quickSort**(…) method.

1a) with median-of-three pivot

1b) with random pivot

1c) with pivot = array[0]

Task 2: **insertionSort**(…) method.

Task 3: **mergeSort**(…) method.

Each method shall use the algorithm to sort a given array of numbers. Note, that the algorithm must be the one described in a book (see [1]). Each method must be implemented as:

a) an **iterative** method and

b) a **recursive** method.

After completing the implementation of each algorithm the running time should be measured, i.e. how long time it takes for each method to sort three arrays containing 100, 10000 and 1000000 numbers. The arrays shall be filled with random numbers read from the file (Seminar1 → File with random numbers).

The output (for each task) should be

- presented in a table, e.g.

Method \ Input	100	1000	1000000
quickSort(…) method, **recursive**			
median-of-three pivot			
random pivot			
pivot = array[0]			
quickSort(…) method, **iterative**			
etc.			

- plotted on a diagram, e.g. using Excell or similar plotting tool. Note: one diagram for each task
- and analyzed.

Do not change the input inside the program, i.e. read the input from the keyboard to easy test other inputs!

Or:

Write a script (code) that run all methods with all inputs. Run your script and take a FIKA while the computer will do the job ☺ After fika you should see all results.

What you can do more with the scripts? Write one that repeats your tests many times and measures the average time. It gives you more trustworthy results (it calls research).

By the way – this is a typical scenario for performance testing.

In the analysis part the comparison between the recursive versus iterative method is expected, as well as the analysis WHY the results look different (for each specific input). Present the theoretical analysis of the code using Big Oh notation and compare the theoretical results to your practical experiment, i.e. to the output you provided in the table as in the plot.

Present one more diagram for the outputs from all the methods but only for the input 1000000 and provide the analysis.

Task 4: **binarySearch**(…) method.

Implement a binarySearch(…) method. The method shall use the Binary Search algorithm to find out if a number sent as a parameter to the method exists in an array that is also sent to the method as a parameter. The binarySearch method must be

implemented as a recursive method and shall return true if the number is found in the array and false if it is not found.

During the Seminar –

Each student is responsible for one (random) Task for the examination. The student is responsible for the task, i.e. explains the algorithm, presents a code, discusses the complexity of the code/algorithm, discusses other students' solutions, as well as is responsible for the analysis of the results of the current task.

Thereafter each student will be individually examined by the examiner during the seminar.

References

1. Weiss, Mark. A. (2012), Data structures and algorithm analysis in Java. 3rd edition Harlow, Essex : Pearson. (632 p).