KTH Datavetenskap
och kommunikation

Course PM for DD2458
Spring 2023

Successful problem solving in computer science requires a solid theoretical foundation as well as ability to apply the theory to practical problem solving. The aim of this course is to develop your ability to apply knowledge of algorithms, data structures, and complexity theory to given problems. As a professional it is useful to be able to analyze a problem, judge the efficiency of proposed algorithms, and to implement them quickly and correctly. In this course, you will practice this by solving a number of homework assignments and while working under time constraints during problem solving sessions.

Learning Outcomes

The overarching goal of the course is that the students should be able to use programming as a tool for problem solving, and should be able to apply theoretical knowledge from other computer science courses to solve practical problems. The course has a large focus on going all the way from theory (in the form of algorithm design) to practice (in the form of a working program).

After passing the course the student should be able to:

- use algorithm design methods such as greedy algorithms, dynamic programming, decomposition, and combinatorial search to construct algorithms for solving given problems,
- apply basic algorithms in areas such as graph theory, number theory, and geometry on given problems and adapt them to problem-specific circumstances,
- analyze the efficiency of different algorithms in order to decide which ones are sufficiently efficient in a given context,
- compare different problems with respect to difficulty,
- implement algorithms and data structures given abstract specifications,
- identify bugs in others' solution attempts on a problem,
- communicate with others during problem solving in groups,
- present algorithms, data structures, and problems verbally in a concise and lucid way.

The goals are attained by solving number of homework assignments, implementing a small library of algorithms and data structures, solving problems in groups during problem solving sessions, and by presenting solutions to homework assignments.

Code of Honor

The EECS Code of Honor applies on the course: »https://www.kth.se/en/eecs/utbildning/hederskodex/inledning-1.17237»

Course Registration

You must register on the course Kattis page: »https://kth.kattis.com/courses/DD2458/popup23» (this lets the course responsible see your results so that grades can be computed and reported).

Course Web Page

All information about the course is on the course page: »https://canvas.kth.se/courses/37562»

Help and Guidance

If you are stuck or have questions, you have the following options:

- Canvas discussion forum: »https://canvas.kth.se/courses/37562/discussion_topics».
- Ask at lectures or other events
- E-mail `austrin@kth.se`

Prerequisites

One of the courses DD2350 *Algorithms, Data Structures, and Complexity* or DD2352 *Algoritms and Complexity*, or equivalent.

Course Literature

As main course book this year, we use a (complete) draft of a book called "Principles of Algorithmic Problem Solving" written by Johan Sannemo. A PDF is available on the course Canvas page.

In addition, books from earlier algorithms courses can probably serve as dictionaries of algorithms. For those who don't already have such a book, we recommend *Introduction to Algorithms, Third Edition* by Cormen, Leiserson, Rivest och Stein.

Schedule

We meet physically Wednesdays 10-12. The first hour, 10-11, consists of homework presentations (described below) and the second hour, 11-12, are lectures, with the exception of the first two weeks (in which there are no presentations and we instead have two-hour lectures).

The remaining activities in the course are done online. Every third week there is a problem session (described below) for a full afternoon 13-18, and most other weeks there are lab session where we will do various exercises to practice our problem solving skills.

Kattis

On all programming components of the course (including problem sessions), we use the "Kattis" system (»https://kth.kattis.com/»). If you have not used Kattis before, log in as soon as possible (using your KTH-ID) and familiarize yourself with it. The help pages has a short tutorial (go to the page for your favorite language).

Examination

The course has two modules in Ladok:

LAB1 Problem sessions and algorithm labs, 4.5 hp.

OVN1 Homework and presentations, 4.5 hp.

Both modules are graded (A-E) and the final grade on the course is an aggregation of the grades on the modules, according to the table below. To pass the course, you need to have at least an E on both modules. The grade on each module is based on grade points that are accumulated by solving problems.

Problem Sessions

There are four problem sessions, consisting of a number (normally 6 or 7) problems that are solved in groups of two during 4.5 hours.

The problem sessions are done in groups of two (in exceptional cases three students, in case someone would be left alone).

Of the four problem sessions, the results on your three best are counted. Each solved problem on these sessions gives 3 grade points to the LAB1 module. If the programming contest "KTH Challenge 2023" takes place during the course, then it may also be counted (but solved problems there do not necessarily give 3 grade points).

Algorithm Labs

There are four labs. Each lab consists of implementing algorithms for some well-known problems (e.g string matching, shortest paths in graphs, systems of linear equations, convex hulls, etc).

The labs are done in groups of up to two people.

Each individual algorithm normally gives 1 grade point (in some rare case, 2 points) on LAB1, and each of the four labs can give in total 9 grade points on LAB1. To obtain grade points for a lab, you need to solve the problems in Kattis, and present your solution orally (over Zoom). In addition to Kattis accepting the code, it is required that the lab code is well structured and documented (see course web page for further details about these requirements).

Homework

There are 12 sets of homework (roughly, one per week). Each set consists of 6 problems.

The homeworks must be solved individually and you must not discuss them with other students prior to the deadline.

Problems solved before deadline give 1 grade point to the OVN1 module, and problems solved after deadline give 0.5 grade points. In each homework set, there are two "$\star$" problems that are considered a bit harder (but what is hard is subjective, and sometimes you may find a $\star$-problem easier than a non-$\star$-problem). The $\star$-problems are used to determine if you have reached the threshold for an A grade on the course (see Grade section below).

Homework Presentations

Every student must participate in presentations of homework solutions. After the deadline for each homework, there is a presentation session where the students present solutions.

At the start of the presentation session you indicate ("check") which problems you are prepared to present. You may only present problems that you solved in Kattis, but you do not have to present all problems that you solved in Kattis.

Then the set of students to present the problems are chosen randomly. If a presentation is not properly prepared, the student loses all their "checks" from the session.

After a presentation, we discuss and give comments together.

Each check gives 0.2 grade points to OVN1.

Grades

LAB1 The grade points on LAB1 are converted to a grade according to the following table:

Grade points	$20^{\dagger}$	30	38	46	54
Grade	E	D	C	B	A

† To pass LAB1, it is additionally required that at least 9 of the grade points are contributed from the problem sessions.

OVN1 The grade points on OVN1 are converted to a grade according to the following table:

Grade points	$18^{\dagger}$	28	36	44	$44^{\ddagger}$
Grade	E	D	C	B	A

† To pass OVN1, it is additionally required that:

- You solved at least one problem per homework set (problems solved after deadline are counted).
- You accumulated at least 3 grade points from checks (i.e., you checked at least 15 times).

‡ To get grade A on OVN1, it is additionaly required that:

- You solved at least one $\star$-problem per homework set ($\star$-problems solved after deadline are counted towards this), and
- You have checked at least 9 $\star$-problems during the course.

If these requirements are not met, the grade on OVN1 is at most B, regardless of how many grade points are attained.

Final Grade Provided that you get a passing grade on both LAB1 and OVN1, the final grade on the course is given by the following table (average grade rounded down):

LAB1 OVN1	A	B	C	D	E
A	A	B	B	C	C
B	B	B	C	C	D
C	B	C	C	D	D
D	C	C	D	D	E
E	C	D	D	E	E

Advice From Previous Years' Students

Here are some responses from previous years' students to the question "What advice would you like to give to future course participants?":

> "Start early, keep going all the way through. Go to bed in time on Tuesdays so that you don't accidentally sleep through the presentation & lecture. Those 'x'es are important if you want to go for a high grade."

> "Do as many homeworks as you can at the start of the course. This will offer you more freedom later."

> "Everyone from last years evaluation was 100% right and I should've listened. To summarize: Decide on what grade you want as early as possible! This course takes a significant amount of time & work! Do the labs, they will come in handy! And the most important piece of advice: don't take robotics at the same time, trust me on this! This will prepare your for programming interview questions like no other course at KTH. Have fun!"

> "If you're like me, and after 4 years at KTH still start with tasks at the last minute, then this course is great. You will definitely learn not to do that any more."

> " The course starts 'slow', I advise you to do as many assignments you can the first two weeks and 'sign up' for showing the solution. The assignments gets harder for each week. "

> " It is very helpful to implement as many lab assignments as possible, since they cover much of the functionality needed in both homework assignments and live programming sessions. So don't think of them as separate point-awarding work, rather think of them as getting twice the points for the same work only because you started early. "

> " Decide the first week what grade you would like and aim for that grade from the start. It is hard to later on decide that you would like a higher grade. Try to take advantage of the weeks where there are easier problems and build yourself a buffer for later. "