Course
CS-E3190
Course materials
Your points

This course has already ended.
The latest instance of the course can be found at: Principles of Algorithmic Techniques: 2023 Autumn

« Kick-Off Lecture Course materials Warm-up to Programming Exercises »
CS-E3190 / Kick-Off Lecture / 1.1 Materials

Materials

This chapter contains the lecture slides for the kick-off lecture. Please consult the slides for all the relevant course arrangements.

Kick-off lecture slides: Welcome.pdf

The key course information is also summarized below for your convenience.

CS-E3190 Principles of Algorithmic Techniques

An **algorithm** is a finite sequence of elementary instructions for accomplishing a well-defined task. A single CPU core today can easily execute billions of elementary instructions per second, which makes efficient algorithms both immensely powerful tools in practice and a subject of considerable mathematical intrigue: Which well-defined tasks can be solved efficiently? What is the most efficient algorithm to accomplish a particular task?

This course will advance your understanding in the basic principles and fundamental challenges in computing. We analyse problems, where we can rigorously argue that a certain algorithm design is good (or bad). The focus is on proving mathematically that certain approaches do (or do not) work. Furthermore, we learn mathematical principles that form the basis for a large number of algorithm designs and their analysis, as well as in select cases turn these into concrete algorithm implementations and measure their performance.

There are no formal pre-requisites for the course, but basic knowledge in discrete mathematics as well as basic programming skills are essential. The course is suitable for students at all levels (bachelor, master, and doctoral) who are interested in expanding their knowledge in algorithms.

Course staff

Lecturer
Jara Uitto http://jarauitto.com

Teaching assistants
Melanie Cambus (head TA)
Etna Lindy
Shreyas Pai
Hossein Vahidi
Mihai Macarie (C++ assistance)

Course material and submissions

The material of the course is based here in the A+ system. Here you can find the exercises, the links to the video lectures and announcements. Furthermore, the submissions of all exercises is done through A+.

Learning objectives

1. **Modeling**: The students are able to derive mathematically well-defined problems from a given real-life problem.
2. **Design**: The students are able to design algorithms using basic algorithmic tools.
3. **Analysis**: The students are able to rigorously analyse the correctness and runtime of an algorithm.
4. **Implementation**: The students are able to implement fundamental algorithms. The main focus of the course is in the first 3 items of this list. However, in the ideal case, a beautiful algorithm design is complemented by a great implementation.

Sessions and the online implementation

Kick-off lecture: On the 9th of September at 14:15 - 16:00, there is a kick-off (live) lecture T1. No exercises sessions on the first week (7th and 8th).

Recorded lectures: Besides the kick-off lecture, all lectures will be recorded videos. The links to the videos will be shared in the beginning of the corresponding week. For questions related to the lectures and the corresponding exercises, we will create a Zulip channel where students can ask questions related to the week. During the Friday sessions (T1 at 14:15 - 16:00) there is also a chance to ask questions related to the lecture videos.

Types of exercises: There are three different types of exercises.

1. **Tutorial exercises**: For each topic, we will have several tutorial exercises that help students to better understand the topic. They are not a prerequisite but are intended to support in solving the graded exercises and the programming exercises. These exercises are discussed weekly in the exercise sessions.
2. **Graded exercises**: For each topic, there is one graded exercise (potentially with a few sub-tasks). In total, there are ten graded exercise sheets. These exercises can be found in and are submitted through the A+ system. The deadlines are roughly one week after the material for the corresponding topic is published and we reserve two weeks for performing the grading. All submissions must be done in PDF form. We recommend using LaTeX. Handwritten solutions (in PDF) are accepted but in case we are unable to read your handwriting, we will grant 0 points. One of the graded exercises per week is an **individual** exercise. The teaching staff will not assist with the **individual** exercises. You are allowed to discuss those with your peers as long as you follow the code of conduct.
3. **Programming exercises**: There are 4 programming exercises, that can be found in and submitted through the A+ system. The programming language is C++. The exercises are automatically graded based on the correctness and efficiency of your solution.

Solutions to the graded exercises: We will not provide model solutions to the graded exercises. On the Friday sessions (14:15 - 16:00 in T1), we will show the solutions to the previous weeks exercises. Furthermore, we give some tips on how to the approach the next round of graded exercises.

Late submissions: Late submissions, to graded and programming exercises, will result in 0 points.

Communication through Zulip: To create an account in our Zulip chat, please follow this link. Just enter your `aalto.fi` email address, and follow the instructions.

We will have a separate channel for each exercise session, for each programming exercise, and for each lecture. The corresponding channel will focus on the exercise at hand. The channels for the lectures act as a general forum for questions and answers. There are two special channels:

#general: Is intended for any general discussion related to the course.
#queue: During the exercise sessions, you can request for 1-on-1 help here.

Each teaching assistant and the responsible teacher will have an account in Zulip. However, do not post direct messages to us, unless you have requested to do so, e.g., during 1-on-1 help during an exercise session.

Working on the exercises (and on this course)

A big part of this course is to obtain individual skills in algorithm design, analysis, and implementation. Hence, everyone should write their solutions to all of the exercises themselves. Please see the Code of Conduct for this course for more details.

The code is by no means intended to forbid working with your peers. On the contrary, we want to encourage working with other students and peer support. To make this explicit, we offer a small amount of extra points for each student that provided help to a peer in some exercise. This will not be controlled, we hope that it simply makes it explicit that helping others is not a bad thing. Just make sure to follow the code of conduct.

Solutions to graded exercises and Q&A (Fridays at 14:15 - 16:00)

During the Friday sessions, we will discuss solutions to the graded exercises. Written model solutions will not be provided. We will also give tips on how to approach the next round of graded exercises. Furthermore, you have the chance to ask questions related to the lecture videos.

Exercise sessions

There are two (identical) exercise sessions each week (see weekly schedule) in T3. In these sessions,

1. you are given time to solve, together or independently, the tutorial exercises and get deeper into the topic.
2. towards the end of the session, the TA will provide solutions to the tutorial exercises.
3. you can work on the graded exercise. Upon request, the TA will give help in solving these exercises.

However, the individual homework assignments will not be discussed in these sessions by the TA.

There are two ways to obtain help:

- **Teaching Staff**: Come to the exercise sessions!
- **Zulip**: You can communicate with a TA using direct messages in Zulip.

In the **#queue** channel in Zulip, you can initiate a help request during an exercise session. In your request, you should give a short description of your issue (below some more instructions). Once there is a TA available, your request will be marked and the TA will contact you.

Help through Zulip:

Go to the #queue channel in Zulip. Write a help request message that starts with the keyword "zulip". Your help request should consist of a few sentences that describe what is the problem, what you know, and what you have already tried. You can also make a pointer (e.g., link) to a relevant part of the course material. A TA will contact you through direct messages in Zulip and assist you with your problem.

Help with programming exercises. The programming exercises are independent work and the teaching staff will not solve these together with the students. However, questions and help can be requested from fellow students and the teaching staff through Zulip.

Grading

The course grading is based on three components:

1. **Graded homework**: Each graded homework is worth 10 points, 90 points in total.
2. **Programming exercises**: Each programming exercise is worth 10 points, 40 points in total. There is a warm-up programming exercise worth 3 points.
3. **Extra points**: Answering each feedback, one per lecture, gives 9 points and exercise help gives 2, in total 11 points.
4. **Total points**: 144

The final course grade is determined based on the total points earned as follows:

Grade 5: At least 107 points.
Grade 4: At least 97 points.
Grade 3: At least 87 points.
Grade 2: At least 77 points.
Grade 1: At least 67 points.

Weekly schedule

	Monday	Tuesday	Wednesday	Thursday	Friday
Lecture published		12:00			
Exercise sessions			14:15–16:00 T3	14:15–16:00 T3	No session
Graded solutions					14:15–16:00

All deadlines are on Mondays at 20:00.

Course calendar and deadlines

All deadlines are on Mondays at 20:00.

Week 36 5.9.–9.9.
Kick-off lecture Friday 9.9. 14:15
No tutorial session this week

Week 37 12.9.–16.9.
Lecture 1
Tutorial session 1
Deadline for programming warm-up

Week 38 19.9.–23.9.
Lecture 2
Tutorial session 2
Deadline graded exercise 1

Week 39 26.9.–30.9.
Lecture 3
Tutorial session 3
Deadline graded exercise 2
Deadline programming exercise 1

Week 40 3.10.–7.10.
Lecture 4
Tutorial session 4
Deadline graded exercise 3

Week 41 10.10.–14.10.
Lecture 5
Tutorial exercise 5
Deadline graded exercise 4
Deadline programming exercise 2

Week 42 17.10.–21.10.
Evaluation week, no activity at this course

Week 43 24.10.–28.10.
Lecture 6
Tutorial session 6
Deadline graded exercise 5

Week 44 31.10.–4.11.
Lecture 7
Tutorial session 7
Deadline graded exercise 6
Deadline programming exercise 3

Week 45 7.11.–11.11.
No new lecture. More time for programming
No tutorial session
Deadline graded exercise 7
Deadline programming exercise 3

Week 46 14.11.–18.11.
Lecture 8
Tutorial session 8
No graded deadline

Week 47 21.11.–25.11.
Lecture 9
Tutorial session 9
Deadline graded exercise 8

Week 48 28.11.–2.12.
Deadline graded exercise 9
Deadline programming exercise 4