Algorithms and Data Structures II (1DL231)
00: Logistics

Justin Pearson
`http://user.it.uu.se/~justin/`

2024

The most important links

- All assignments will be handled by Studium.
- Information on lectures, assignments and the exam can be found at

  `https://ad2-uu-se.github.io/`

There is also a link via Studium.

You should spend some time reading the webpage. There are answers to many of your questions there.

Announcements will be made via Studium.

Guest Lectures 2023

- Frej Knutar Lewander will give a lecture on graph algorithms and the assignments' API.
- Pierre Flener will give the two lectures on P & NP.
- He will also answer questions and talk a bit about the followup courses to this course: AD3 and Combinatorial modelling.
- You will be examined on this material.

Learning Outcomes, from the Catalogue

In order to pass, the student must be able to:

- use the notation of asymptotic growth of functions and be able to use this notation to describe the complexity of algorithms and computational problems;
- derive equations for the complexity of algorithms and solve such equations;
- work with common algorithmic techniques such as dynamic programming, greedy algorithms, etc;
- deal with basic problems using graph algorithms, string matching, and flow networks;
- define the complexity classes P and NP, and discuss the open question whether P = NP;
- present & discuss topics related to the course content orally and in writing with a skill appropriate for the level of education.

  ☞ written reports & oral resubmission!

Course Content

- Algorithm analysis
- Dynamic programming
- Revision on Graph algorithms
- Greedy Algorithms, Minimal spanning trees and Prim's algorithm
- Network flows
- Union find data structure
- P vs NP and an introduction to complexity analysis.

Course Organisation & Time Budget

Period 2: - budget = 133.3 h:

- 13 lectures, taught in English: budget = 19.5 hours
- 3 assignments with 3 help sessions, 1 grading session, and 1 solution session per assignment, on 2 problems each, to be done by the student :
  budget = avg 30 hours / assignment / student (2 credits)
  counting also as exam preparation!
- 1 written closed-book multiple-choice exam
  to be done alone: further budget = 24 hours (3 credits)
- Prerequisites: 60 c, including minimum 15 c in maths and 30 c in computer science, including Algorithms and Data Structures 1 (1DL210) or equivalent

Assignments

- The assignments are to be completed with the aid of LLMs or other AI. So no ChatGPT, no Copilot, no Claude or whatever.
- The code should be completely your own work.
- If you use AI/LLMs, then this is considered cheating and will be reported.

Why are we doing this? Why should I learn this stuff if an LLM can do it?

The only way to master this material is to learn how to implement and reason about the algorithms yourself. You have to start with the easy problems before you move on to the more complicated problems. To be able to review what comes out of an LLM you have to be able to understand the code.

Assignments

- Three assignments based on the course material.
- Working on the assignments is also excellent revision for the exam.
- Assignments are solo assignments. For each assignment there are optional parts that you can do if you aim for a higher grade. The grading rules are published on the web.[1]
- Only aiming for a 3 is very risky. The early assignments are easier than the later assignments. Even if you are aiming for a 3 on the assignments it is worth going full blast on the first assignment.
- There is a minimum threshold 3 points on each assignment. Each assignment consists of two problems. On each problem you need to get at least 1 point. This means that you have to attempt both problems.

[1] `https://ad2-uu-se.github.io/grades/index.html`

Help Sessions and Solution Sessions

- Help sessions are the only way to get help on your assignments. We will not answer questions via email, and there is no forum for discussion on Studium (I had one in previous years and nobody used it).
- Solution sessions are when the solutions are published. We go through common mistakes and problems students have. This is instead of having more detailed feedback on your assignments (this is impossible with so many students). We do not publish any solutions to the assignments.

Help sessions are not lab sessions. You are expected to work on the assignment outside the help sessions and come to the help sessions with questions.

Assignment Grade and Deadlines

All assignments are published before the course starts.

- submission, by 13:00 Swedish time. This deadline is strict. Grading starts at 13:00.
- Final results within 10 working days. For assignment 3, latest three days before the exam.

Assignment 1 has a grading session where students who do not have a passing grade (but nonzero scores) are given the chance to increase their score. Typically students make silly mistakes that can be fixed easily.

For all other assignments there is no solution session and no chance to resubmit while the course is running.

Assignments 1 and 2, have a solution session where we present common problems that students have with their assignments. There is limited marking time, and we do not have time to give detailed feedback on each report.

Expectations on the Assignments

Passing the given test cases is only the start.

- Your code must pass all provided test cases. If you do not pass all the test cases then we will not look at your report and you will get a score of 0.
- You have to develop a correct implementation of the problem. Read the specification in the assignment carefully.
- The written report is a large part of the mark of the assignment. The writing is one of the course goals of the project. In the report you have to argue for the correctness and complexity of your solution.
- Please look at the demo report.

If you do not pass all the provided test cases then you will get 0, and we will not look at the report. You are supposed to provide a correct implementation. The test cases are there to help you write a correct implementation.

When handing in your assignment

- Read the checklist for your report. It is the first page of the demo report.
- Follow the coding convention. Use the given APIs, again this allows us to test your code. If you change the API, then you will get 0 points.
- Hand in both the report and your code. If you forget to hand in the code, then it is automatic failure.

Do not ask for extensions or exceptions. The marking scheme is very tight. With the volume of students we cannot deal with exceptions. Plan ahead. All the assignments are online, you can start now.

The assignment should be completely your own work, no plagiarism and no AI assistance. The only way to learn this material is to engage with it yourself.

2 Assignment Credits & Overall Influence

Let a_{ij} be final score on Problem j of Assignment $i \in 1..3$:

- 20% threshold: $\forall i, j : a_{ij} \geq 20\% \cdot 5 = 1$
  No catastrophic failure on individual problems
- 30% threshold: $\forall i : a_i = a_{i1} + a_{i2} \geq 30\% \cdot (5 + 5) = 3$
  Can offset partial failure on problems or assignments
- 50% threshold: $a = a_1 + a_2 + a_3 \geq 50\% \cdot 3 \cdot 2 \cdot 5 = 15$
  The formula for grades 3..5 is at the course homepage
- Worth going full-blast: The assignment score a is meshed with the exam score e in order to determine the overall course grade in 3..5, **if** $15 \leq a \leq 30$ **and** $15 \leq e \leq 30$: see the formula at the course homepage.

For each problem, Python skeleton code with extensive unit test cases are provided. Your code must pass all provided test cases. If you do not pass all the test cases then we will not look at your report and you will get a score of 0.

How To Communicate by Email?

To email the assistants or the head teacher:

- If you have a question about the **lecture material** or **course organisation**, then contact the head teacher.
- If you have a question about the **assignments** or **infrastructure**, then contact the assistants at a help or solution session for an immediate answer.
  Short *clarification* questions (that is: *not* about programming issues) that are emailed to it-ad2@lists.uu.se are answered as soon as possible during working days and hours.
  No answer means that you should go to a help session: almost all the assistants' budgeted time is allocated to grading and to the help, grading, and solution sessions.

Please do not use Studium to contact me, but rather use it-ad2@lists.uu.se, which goes to all of the assistants and me. If for some reason you want the email to only go to me then use justin.pearson@it.uu.se.

What Has Changed in recent years?

Recent changes triggered by the previous course evaluations:

- Assignments are now solo. One of the biggest complaints and problems is that people have trouble working in pairs or finding partners. The assignments optional parts that you can do to get a higher grade.
- The sanity-check server for assignments was retired, due to hardware unreliability and lack of resources for improvements.
- Improved instructions for the assignments.
- More test cases for you to run for the assignments. Passing the test cases is only the beginning, but if you code does not pass the test case, then you should think why. We have tried to find both trivial sanity checking test cases, and test cases that try to find corner cases in the code.
- We removed a revision lecture of AD1 material. Many students complained that it was not necessary. The only revision of AD1 material is today.