Course Introduction

This course, taught by Rolf Fagerberg from the Department of Mathematics and Computer Science (IMADA), focuses on algorithms and data structures.

Participants

The course is designed for a diverse group of students, including:

- BA in Computer Science (2nd semester)
- BA in Artificial Intelligence (2nd semester)
- BA in Software Engineering (4th semester)
- BA in Mathematics-Economics (4th semester)
- BA in Applied Mathematics (4th semester)
- BA minor in Computer Science (6th semester)
- MA profile in Data Science (2nd/8th semester)

Course Structure

The course is offered under different codes, each with a slightly different structure:

- DM578: Algorithms and data structures (7.5 ECTS), assessed via a 3-hour written multiple-choice exam.
- DM507/DS814: Algorithms and data structures (7.5 ECTS) plus a programming project in three parts (2.5 ECTS), and a 3-hour written multiple-choice exam.
- SE4-DMAD: Algorithms and data structures (7.5 ECTS) combined with discrete mathematics (2.5 ECTS), assessed via a 3.75-hour written multiple-choice exam. Note that discrete mathematics has separate lectures (by Lene Monrad Favrholdt) and exercise sessions.

There are separate ITS course rooms for DM507/DM578/DS814 and SE4-DMAD.

Course Format

The format for DM578, DM507/DS814, and SE4-DMAD includes lectures by Rolf Fagerberg and Lene Monrad Favrholdt, exercise sessions led by an instructor, and both independent and group work.

Prerequisites: Programming experience in Python or Java and some mathematical maturity.

Examination

The examination for all course codes is a written multiple-choice exam in June. The exams are open book and notes and are graded on the 7-point scale. The goal is to check the students' knowledge of the material. The re-examination is oral.

DM507/DS814 also includes a programming project in Python, worth 2.5 ECTS, graded as Passed/Not Passed. Passing the project is not required to take the written exam, and is aimed to train the transfer of the material to practice (programming).

Course Materials

- Textbook: Cormen, Leiserson, Rivest, Stein: Introduction to Algorithms, 4th edition, 2022. (Reading material and assignments will also be given from the 3rd edition.)
- Lecture slides
- Exercises for exercise sessions
- Past exam papers
- Project (for DM507/DS814)
- Other learning materials on the course website

Expected Workload

The expected workload varies slightly depending on the course code:

Task	DM578	DM507/DS814	SE4-DMAD
Skimming before lecture	0.5 hrs	0.5 hrs	0.5 hrs
Lecture	2 hrs	2 hrs	2 hrs
Reading after lecture	1.5 hrs	1.5 hrs	1.5 hrs
Homework exercises	3 hrs	3 hrs	3 hrs
Class exercises	2 hrs	2 hrs	2 hrs
Project	-	55 hrs	-
Exam reading	40 hrs	40 hrs	40 hrs
Question time and exam	6 hrs	6 hrs	6 hrs
Total estimated hours	235 hrs	290 hrs	280 hrs
ECTS	7.5	10	10

Note that for SE4-DMAD, the workload is calculated with 1.5 sessions per week over 14 weeks (Rolf) and 0.5 sessions per week over 10 weeks (Lene).

Course Objectives

The course aims to enable computers to perform tasks efficiently, touching upon:

- **How to write programs**: Programming, programming languages, software engineering.
- **How the program should solve the task**: Algorithms and data structures, database systems, linear algebra with applications, data mining, and machine learning.
- **The limits of problem-solving**: Lower bounds, complexity, computability.
- **How the machine executes tasks**: Background knowledge of computer architecture and operating systems.

Problem-Solving Approach

Algorithms

> An algorithm is a solution method, precisely described through drawings, text, or pseudo-code.

Data Structures

A data structure consists of data and efficient operations on that data.

Levels of Solution

1. Invent an algorithm to solve the task.
2. Compare several algorithms that solve the task.
3. Determine the best possible algorithm to solve the task.

Algorithm Development and Evaluation

1. **Invention**: Requires ideas, experience, and a toolkit of existing algorithms and development methods.
2. **Comparison**: Requires defining quality (often measured by low time consumption).
3. **Optimality**: Finding the best possible algorithm.

Analysis (thought work, arguments, proofs) is a valuable tool for points 1, 2, and 3. It provides high certainty of the method's correctness, saves implementation work, and offers comparisons unaffected by machine, language, programmer, or test data selection.

Testing (implementation, testing) is a good tool for points 1 and 2. It can explore ideas, catch implementation errors, and highlight aspects not captured by analysis.

DM507/DM578/DS814/SE4-DMAD will focus most on analysis, with less emphasis on implementation and testing.

Course Goals

The course aims to equip students with a toolkit of algorithms for fundamental tasks and methods to develop and analyze new algorithms and variants of existing ones. Exercises and programming projects enhance understanding and train the use of these tools.

Specific Course Content

- Algorithms:
  - Analysis of algorithms: correctness and runtime (tool)
  - Divide and conquer algorithms (algorithm method)
  - Greedy algorithms (algorithm method)
  - Dynamic programming (algorithm method)
  - Specific algorithms for sorting, graph problems (BFS, DFS, shortest paths, spanning trees), file compression, matrix multiplication, etc.
- Data structures:
  - Dictionaries (search trees and hashing)
  - Priority queues (heaps)
  - Disjoint sets