

Introduction to Algorithms and Data Structures, Course Description

Why this course?

When developing software it is important to know how to solve problems in a computationally efficient way. Algorithms describe methods for solving problems under the constraints of the computers resources. Often the goal is to compute a solution as fast as possible, using as few resources as possible. To solve a problem efficiently it may be necessary to use data structures tailored for the particular problem(s) at hand. A data structure is a specific way of organizing data that supports efficient performance of the relevant operations on that data. For instance there are data structures for organizing large numbers of records where records already present can be quickly found and/or deleted, and new records can be inserted and found fast.

Goal

The goal of the course is to give a basic understanding of how common computational problems can be solved efficiently on a computer.

The goal of the course is to give the student a basic understanding of fundamental algorithms and data structures so that the student:

- **Is able to describe the basic data structures, algorithms and programming techniques**, including lists, stack, queues, search trees, hash tables, different sorting algorithms, search algorithms for graphs, and dynamic programming.
- **Can apply the techniques from the course when solving programming/algorithm problems.** These techniques include methods for sorting and searching, basic graph algorithms, recursion, dynamic programming, and time and space analysis of programs, both basic techniques and amortized analysis.
- **Is able to select the best algorithm and/or data structure when solving a given programming problem.**
- **Is able to analyse time and space required for the execution of a program, as well as the correctness of a program.**
- **Is able to formulate a given programming task as an algorithmic problem, in order to select the best method for solving it.**
- **Is able to combine and modify algorithms and data structures, in order to design an efficient program.**

All in all this course teaches you how to write fast programs, how to design them, and how to analyse them.

After the course the student has the algorithmic competences required to meet the prerequisites for advanced courses such as "Advanced Algorithms", "Programming Languages, Interpreters, and Compilers", and "Advanced Database Technology".

Contents

This course is a "must have"-course, for anyone who wants to become a good programmer. The course will provide the students with a toolbox of highly relevant techniques for solving common, as well as many specialized, programming problems.

The course takes its cue from different problems, which will be solved by using selected basic algorithm/advanced programming techniques. Subjects which we will cover:

- Methods to sort and search, including Merge sort, Quick sort, balanced search trees, and hashing.
- Methods to find a shortest path in a graph, such as Bellman-Ford and Dijkstras algorithms.
- Methods to store large data sets for efficient access and manipulation, e.g. Union-Find data structures.
- Methods to analyse how time/space efficient a program is, including Big-Oh analysis.
- Methods to use recursion in programming.
- Programming methods like dynamic programming.

This will include e.g.:

- Stacks, queues, sequence, trees.
- Priority queues, search trees, dictionaries.
- Graph algorithms.

We will also look at different tools to analyse programs, such as invariants to prove correctness, and asymptotic analysis to calculate time and space needed for a program.

Textbook

1. T.H.Cormen, C.E.Leiserson, R.L.Rivest, C.Stein. Introduction to Algorithms. 2nd ed. The MIT Press (1998). ISBN 0-262-53196-8.
2. Robert L. Kruse, Data Structure and Program Design, PHI
3. Heileman :Datastructure Algorithms &OOP
4. Data Structure Using C – M.Radhakrishnan, V.Srinivasan, ISTE/EXCEL BOOKS
5. Weiss Mark Allen, “Algorithms, Data Structures, and Problem Solving with C++”, Addison Wesley.
6. Horowitz Ellis & Sartaj Sahni, “Fundamentals of Data Structures”, Galgotria Pub.
7. Tanenbaum A. S. , “Data Structures using ‘C’ ”

Language

The course is taught in English.

Programming language

The course will contain a number of programming assignments, which you can choose to solve. In principle any imperative programming language can be used to solve the assignments. Help can primarily be given in C.

Examination

Surprise Test

Class Test

Midterm Examination

Final Examination