CS-250

Algorithms I

Chiesa Alessandro, Svensson Ola Nils Anders

Cursus	Sem.	Type
Communication systems minor	E	Opt.
Communication systems	BA4	Obl.
Computational science and Engineering	MA2, MA4	Opt.
Computational science and engineering minor	E	Opt.
Computer science minor	E	Opt.
Computer science	BA4	Obl.
Cyber security minor	E	Opt.
Data science minor	E	Opt.
HES - IC	E	Obl.
Mathematics	BA6	Opt.

Language of teaching	English
Credits	8
Session	Summer
Semester	Spring
Exam	Written
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 weekly
Number of positions	

Summary

The students learn the theory and practice of basic concepts and techniques in algorithms. The course covers mathematical induction, techniques for analyzing algorithms, elementary data structures, major algorithmic paradigms such as dynamic programming, sorting and searching, and graph algorithms.

Content

Mathematical Induction

• Mathematical background, Euler's formula for trees.

Analysis of Algorithms

• O-notation, time and space complexity, recurrence relations, probabilistic analysis.

Data structures

• Arrays, linkes lists, trees, heaps, hashing, graphs.

Design of algorithms by induction

• Divide-and-conquer algorithms, dynamic programming.

Greedy Algorithms

• Spanning tree and shortest path algorithms.

Sorting and searching

• **merge sort, bucket sort, quicksort, heapsort, binary search.**

Graphs algorithms and data structures

• Graphs traversals, shortes path, spanning trees, transitive closures, decompositions, matching, network flows.

Keywords

Algoriths, data structures, efficiency, problem solving

Learning Prerequisites

Recommended courses

CS-101 Advanced ICC I

Learning Outcomes

By the end of the course, the student must be able to:

- Illustrate the execution of algorithms on example inputs
- Describe basic data structures such as arrays, lists, stacks, queues, binary, search trees, heapas, and hash tables
- Analyze algorithm efficiency
- Compare alternative algorithms and data structures with respect to efficiency
- Choose which algorithm or data structure to use in different scenarios
- Use algorithms and data structures taught in the course on concrete problem instances
- Design new algorithms and data structures bases on known methods
- Prove the corrrectness of an algorithm

Teaching methods

Ex cathedra lecture, exercises in classroom

Assessment methods

Continous assessment with final exam.

Resources

Moodle Link

- https://go.epfl.ch/CS-250