Advanced algorithms

CS-450

Advanced algorithms

Svensson Ola Nils Anders

Cursus

Cyber security minor  
Data Science  
Informatique et communications  
Informatique  
Mineur en Data science  
Mineur en Informatique  
SC master EPFL  
Science et ing. computationelles

Sem.  
E  
MA2, MA4  
MA2  
E  
E  
MA2, MA4  
MA2, MA4

Type  
E Opt.  
Obl.  
Obl.  
Obl.  
E Opt.  
E Opt.  
MA2, MA4  
MA2, MA4

Language  
English

Credits  
7

Session  
Summer

Semester  
Spring

Exam  
Written

Workload  
210h

Weeks  
14

Hours  
7 weekly

Lecture  
4 weekly

Exercises  
2 weekly

Project  
1 weekly

Number of positions

Summary

A first graduate course in algorithms, this course assumes minimal background, but moves rapidly. The objective is to learn the main techniques of algorithm analysis and design, while building a repertory of basic algorithmic solutions to problems in many domains.

Content


Keywords

See content.

Learning Prerequisites

Required courses

An undergraduate course in Discrete Structures / Discrete Mathematics, covering formal notation (sets, propositional logic, quantifiers), proof methods (derivation, contradiction, induction), enumeration of choices and other basic combinatorial techniques, graphs and simple results on graphs (cycles, paths, spanning trees, cliques, coloring, etc.).

Recommended courses

An undergraduate course in Data Structures and Algorithms.
An undergraduate course in Probability and Statistics.

Important concepts to start the course

Basic data structures (arrays, lists, stacks, queues, trees) and algorithms (binary search; sorting; graph connectivity); basic discrete mathematics (proof methods, induction, enumeration and counting, graphs); elementary probability and statistics (random variables, distributions, independence, conditional probabilities); data abstraction.

Learning Outcomes

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

• Use a suitable analysis method for any given algorithm
• Prove correctness and running-time bounds
• Design new algorithms for variations of problems studied in class
• Select appropriately an algorithmic paradigm for the problem at hand
• Define formally an algorithmic problem

Teaching methods
Ex cathedra lecture, reading

Assessment methods

Supervision
Office hours Yes
Assistants Yes
Forum Yes
Others For details, see the course web page.

Resources

Bibliography
See web page for the course.

Ressources en bibliothèque
• Randomized Algorithms / Motwani
• Approximation Algorithms / Vazirani
• Computational Complexity / Papadimitrou
• Algebraic Complexity Theory / Bürgisser
• Quantum Computation and Quantum Information / Nielsen

Notes/Handbook
Class notes and references for the running semester will be provided as needed within a few days after each lecture.

Websites
• http://theory.epfl.ch/courses/AdvAlg/