

CS-450

**Algorithms II**

Kapralov Michael, Svensson Ola Nils Anders

<b>Cursus</b>	<b>Sem.</b>	<b>Type</b>
Computational science and Engineering	MA1, MA3	Opt.
Computational science and engineering minor	H	Opt.
Computer and Communication Sciences		Opt.
Computer science minor	H	Opt.
Computer science	MA1, MA3	Obl.
Cybersecurity	MA1, MA3	Obl.
Data Science	MA1, MA3	Obl.
Quantum Science and Engineering	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Obl.
Statistics	MA1, MA3	Opt.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
<b>Hours</b>	<b>7 weekly</b>
Courses	4 weekly
Exercises	3 weekly
<b>Number of positions</b>	

**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**

Algorithm analysis techniques: worst-case and amortized, average-case, randomized, competitive, approximation. Basic algorithm design techniques: greedy, iterative, incremental, divide-and-conquer, dynamic programming, randomization, linear programming. Examples from graph theory, linear algebra, geometry, operations research, and finance.

**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

- midterm (30%)
- homework (30%)
- final exam (40%)

### Supervision

Forum	Yes
Others	For details, see the course web page

### Resources

#### Websites

- <http://theory.epfl.ch/courses/AdvAlg/>

#### Moodle Link

- <https://go.epfl.ch/CS-450>