CS-450  Advanced algorithms  
Kapralov Mikhail

<table>
<thead>
<tr>
<th>Cursus</th>
<th>Sem.</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyber security minor</td>
<td></td>
<td>E Opt.</td>
</tr>
<tr>
<td>Cybersecurity</td>
<td>MA2, MA4</td>
<td>Obl.</td>
</tr>
<tr>
<td>Data Science</td>
<td>MA2, MA4</td>
<td>Obl.</td>
</tr>
<tr>
<td>Informatique et communications</td>
<td></td>
<td>Obl.</td>
</tr>
<tr>
<td>Informatique</td>
<td>MA2, MA4</td>
<td>Obl.</td>
</tr>
<tr>
<td>Mineur en Data science</td>
<td></td>
<td>E Opt.</td>
</tr>
<tr>
<td>Mineur en Informatique</td>
<td></td>
<td>E Opt.</td>
</tr>
<tr>
<td>SC master EPFL</td>
<td>MA2, MA4</td>
<td>Opt.</td>
</tr>
<tr>
<td>Science et ing. computationelles</td>
<td>MA2, MA4</td>
<td>Opt.</td>
</tr>
</tbody>
</table>

**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
• Quantum Computation and Quantum Information / Nielsen
• Algebraic Complexity Theory / Buergisser
• Computational Complexity / Papadimitrou

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/