

CS-455

Topics in theoretical computer science

Svensson Ola Nils Anders

Cursus	Sem.	Type
Computer science minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	4 weekly
Courses	3 weekly
Exercises	1 weekly
Number of positions	

Remark

Cours biennal

Summary

The students gain an in-depth knowledge of several current and emerging areas of theoretical computer science. The course familiarizes them with advanced techniques, and develops an understanding of fundamental questions that underlie some of the key problems of modern computer science.

Content

This course explores the power of randomness in algorithm design, highlighting how probabilistic techniques can lead to elegant and efficient solutions to a wide range of computational problems. We will cover both foundational methods and advanced applications, with topics drawn from graph algorithms, data structures, learning, and more.

The goal is to introduce core probabilistic tools through concrete algorithmic applications. Topics may include: algorithms for finding min-cuts (including isolating cuts and near-linear time methods), graph sparsification (e.g., Karger's and Benczúr-Karger techniques), balls-and-bins processes and the power of two choices, low-distortion embeddings and dimension reduction, randomized rounding and discrepancy minimization, fast randomized algorithms for approximate counting via Markov chains, learning from samples, differential privacy, and randomized methods in geometry and communication complexity.

We will also discuss fundamental tools such as the Chernoff bound, the Lovász Local Lemma, martingale inequalities, limited independence, and coupling methods. Selected topics such as online learning, bandits, planted and semi-random models, and randomized primality testing may be included depending on time and interest.

Keywords

Randomized Algorithms, Probabilistic Techniques, Graph Algorithms, Sparsification, Approximate Counting

Learning Prerequisites**Required courses**

Bachelor courses on algorithms and discrete mathematics, mathematical maturity.

In particular, we expect students to have a good background in algorithm design, and a solid understanding of probability; we will use some linear algebra and calculus, and mathematical maturity is a must.

Recommended courses

CS-250 Algorithms I, MATH-232 Probability and statistics (for IC)

Learning Outcomes

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

- Design efficient algorithms for variations of problems discussed in class;
- Analyze randomized processes
- Explain basic randomized algorithmic techniques
- Use randomness in algorithm design
- Elaborate on related research questions

Teaching methods

Ex cathedra, homeworks, reading

Expected student activities

Attendance at lectures, completing exercises, reading written material

Assessment methods

- Continuous control

Supervision

Others Electronique forum : Yes

Resources

Bibliography

Randomized Algorithms by Rajeev Motwani and Prabhakar Raghavan
Probability and Computing by Michael Mitzenmacher and Eli Upfal
The Probabilistic Method by Noga Alon and Joel Spencer.
We will post notes and readings from books and research papers.

Ressources en bibliothèque

- [Find the references at the Library](#)

Websites

- <https://theory.epfl.ch/courses/topicstcs/>

Moodle Link

- <https://go.epfl.ch/CS-455>