

CS-448

**Sublinear algorithms for big data analysis**

Cursus	Sem.	Type
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
<b>Hours</b>	<b>3 weekly</b>
Lecture	2 weekly
Exercises	1 weekly
<b>Number of positions</b>	

**Remark**

Pas donné en 2023-24 - Cours biennal, donné les années impaires

**Summary**

In this course we will define rigorous mathematical models for computing on large datasets, cover main algorithmic techniques that have been developed for sublinear (e.g. faster than linear time) data processing. We will also discuss limitations inherent to computing with constrained resources.

**Content**

The tentative list of topics is:

**Streaming:** given a large dataset as a stream, how can we approximate its basic properties using a very small memory footprint? Examples that we will cover include statistical problems such as estimating the number of distinct elements in a stream of data items, finding heavy hitters, frequency moments, as well as graphs problems such as approximating shortest path distances, maximum matchings etc.;

**Sketching:** what can we learn about the input from a few carefully designed measurements (i.e. a 'sketch') of the input, or just a few samples of the input? We will cover several results in sparse recovery and property testing that answer this question for a range of fundamental problems;

**Sublinear runtime:** which problems admit solutions that run faster than it takes to read the entire input? We will cover sublinear time algorithms for graph processing problems, nearest neighbor search and sparse recovery (including Sparse FFT);

**Communication:** how can we design algorithms for modern distributed computation models (e.g. MapReduce) that have low communication requirements? We will discuss graph sketching, a recently developed approach for designing low communication algorithms for processing dynamically changing graphs, as well as other techniques.

**Keywords**

streaming, sketching, sparse recovery, sublinear algorithms

**Learning Prerequisites****Required courses**

Bachelor courses on algorithms, complexity theory, and discrete mathematics

**Important concepts to start the course**

*Discrete probability; mathematical maturity*

**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 space/time/communication complexity of randomized algorithms
- Prove space/time/communication lower bounds for variations of problems discussed in class
- Choose an appropriate algorithmic tool for big data problem at hand

### Teaching methods

Ex cathedra, homeworks, final

### Assessment methods

Continuous control

### Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

### Resources

#### Moodle Link

- <https://go.epfl.ch/CS-448>