Spring 2025
Sublinear Algorithms for Big Data Analysis (CS-448)

Schedule:	Wednesdays 13–16 in CM1120
Professor:	Michael Kapralov <michael.kapralov@epfl.ch>, INJ113
TA:	Kshiteej Sheth <kshiteej.sheth@epfl.ch>, INJ110
Course webpage:	http://theory.epfl.ch/kapralov/courses/cs448
Grades:	homeworks (60%) and final (40%)

Course description

As the sizes of modern datasets grow, many classical polynomial time, and sometimes even linear time, algorithms become prohibitively expensive: the input is often too large to be stored in the memory of a single compute node, is hard to partition among nodes in a cluster to avoid communication bottlenecks, or is very expensive to acquire in the first place. Thus, processing of such datasets requires a new set of algorithmic tools for computing with extremely constrained resources. This course is about *sublinear algorithms*, i.e. algorithms whose resource requirements are substantially smaller than the size of the input that they operate on. We will define rigorous mathematical models for computing with constrained resources, cover main algorithmic techniques that have been developed for sublinear data processing, as well as discuss limitations inherent to computing with constrained resources.

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 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.

Prerequisites

Bachelor courses on algorithms, complexity theory, and discrete mathematics; mathematical maturity.

Grading

The grade will be based on regular homework assignments (60%) and a final (40%). Collaboration on the homeworks is allowed, but each student must write up their own solutions, and list collaborators on the assignment.