### 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
Examples of topics that will be covered include:

- **Laplacians, random walks, graph sparsification:** It is possible to compress graphs while approximately preserving their spectral properties (in particular, properties of random walks)? We will cover the main results from the recent influential line of work on spectral sparsification that provides such compression schemes.

- **Laplacian system solvers:** given a linear system $Ax=b$, how quickly can we find $x$? We will cover nearly linear time algorithms for solving $Ax=b$ when $A$ is a symmetric diagonally dominant matrix (a common scenario in practice) that crucially rely on spectral graph sparsification.

- **Spectral clustering:** given a graph, can we find a partition of the graph into $k$ vertex disjoint parts such that few edges cross from one part to another? This is the fundamental graph clustering problem that arises in many applications. We will cover several results on spectral graph partitioning, where one first embeds vertices of the graph into Euclidean space using the bottom few eigenvectors of the graph Laplacian, and then employs Euclidean clustering primitives to find the partition.

- **Local clustering with random walks:** Given a very large graph and a seed node in it, can we find a small cut that separates the seed node from the rest of the graph, without reading the entire graph? We will cover local clustering algorithms, which identify such cuts in time roughly proportional to the number of vertices on the small side of the cut, by carefully analyzing distributions of random walks in the graph.

### Keywords
- spectral graph theory, sparsification, clustering, random walks

### Learning Prerequisites
**Required courses**
Bachelor courses on algorithms and discrete mathematics, 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 approximation quality of spectral graph algorithms;

Teaching methods
Ex cathedra, homeworks, reading

Expected student activities
Attendance at lectures, completing exercises, reading written material

Assessment methods
• Continuous control

Supervision
Office hours  Yes
Assistants  Yes
Others  Electronique forum : Yes

Resources
Bibliography
There is no textbook for the course. Notes will be posted on the course website.

Ressources en bibliothèque
• Randomized Algorithms / Motwani

Moodle Link
• https://go.epfl.ch/CS-455