UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

SYLLABUS DEL CORSO

Advanced Techniques for Combinatorial Algorithms

1920-87R-04

Aims

This is the first PhD-level course on the design and the analysis of efficient algorithms, with a strong emphasis on theoretical aspects. The prerequisites are a very basic knowledge of graph theory and computational complexity, as well as a good understanding of undergraduate-level courses of algorithms, discrete mathematics, and operation research.
It is not necessary to have already taken a graduate-level course on algorithms.
The goal of the course is to give a broad coverage of the main techniques in algorithm design and analysis, so that the course can be useful also for a researcher in a different field. To this purpose, the computational problems tackled are among the most basic problems on strings and graphs, such as pattern matching, vertex cover and max cut.
In fact, anyone that deals with one of the following questions will find the course interesting.

- How can I cope with problems that are provably hard to solve exactly (i.e. are NP-hard)?
- How can I exploit the massive availability of CPUs?
- How can I query efficiently massive texts?

Contents

Computational approaches: parallel, disk-based, streaming, and moderately exponential-time algorithms

Giving up exact solutions: randomized and approximation algorithms

Text algorithms

Detailed program

1. Parallel algorithms. (PRAM model, prefix sum algorithm, Map-Reduce)
2. Randomized algorithms. (Karp-Rabin algorithm for pattern matching)
3. Text indexing (Suffix trees, suffix arrays, pattern matching, generalized suffix trees, Longest Common Substring between two strings)
4. Fixed-parameter algorithms. (Vertex cover)
5. Approximation algorithms. (Algorithms for Vertex Cover, Max Cut.)
6. Approximation algorithms. (Satisfiability, Approximation Complexity)
7. Text indexing (Computing a suffix array from the suffix tree, Longest Common Prefix LCP array)
8. External-memory algorithms. (Parallel Disk Model, Sorting, B-trees)
9. Streaming algorithms. (Heavy Hitters, Count-Min sketch, Reservoir sampling)
10. Text indexing (Burrows-Wheeler Transform and FM-index)

Prerequisites

- Basic data structures (lists, arrays, queues, hash, search trees)
- Sorting algorithms
- Notions of graphs, trees, connected components
- Turing machines, RAM model
- Linear programming, symplex, polytopes.

All prerequisites can be found in the following books, but the relevant wikipedia pages provide everything we need:

1. Introduction to Algorithms (Cormen et al)
2. Combinatorial Optimization: Algorithm and Complexity (Papadimitriou, Stiglitz)

Teaching form

Lectures

Textbook and teaching resource

Semester

Assessment method

The exam will consists of some exercises that will be given during the course.
Collaboration is encouraged, but each student must write their own version of the solution.

Office hours

on appointment