MC658 - Design and Analysis of Algorithms III

Since 2016

Programme:

1. Classes of Problems

- Complexity hierarchy. Classes P, NP, NP-hard and NP-complete
- Concept of completeness and Cook's Theorem
- Fundamental problems and reductions in NP-completeness
- Other classes of problems: co-NP, PSPACE, undecidable problems (Halting Problem)

2. Exact algorithms

- Pseudo-polynomial algorithm for the Knapsack Problem
- Backtracking Algorithms. Suggested examples:
- Graph coloring
- Subset sum
- Branch-and-bound algorithms: Knapsack problem
- Integer Linear Programming as a tool for solving NP-hard problems

3. Approximation algorithms

- Basic definitions: absolute approximation, approximation factor
- Absolute approximation. Suggested examples:
    - File allocation into 2 disks
    - Coloring of planar graphs
- Inapproximability in absolute approximation. Suggested examples:
    - Knapsack Problem
    - Maximum Clique
- Approximation factor. Suggested examples:
    - Vertex cover problem
    - Metric TSP
    - Set Cover
    - Task Scheduling
- Bin packing
- Inapproximability for approximation factor. Suggested examples:
    - TSP
- Polynomial Time Approximation Scheme. Suggested examples:
    - solving Knapsack Problem by Dynamic Programming
- Use of LP in design of approximation algorithms. Suggested examples:
    - Vertex cover
    - Set cover
    - Max-SAT

4. Heuristics

- Basic definitions: constructive algorithms and local search algorithms
- Greedy constructive algorithms. Suggested examples:
    - Vertex cover
- Local search algorithms. Suggested Examples:
    - 2-opt (or 2-exchange) for TSP
    - Simple exchange for Max Cut
- Meta-heuristics. Suggested examples:
    - GRASP
    - Tabu Search
    - Simulated Annealing
    - Genetic Algorithms