

Advanced Algorithms (Winter 2018/2019)

Instructors:

László Kozma (lectures), Katharina Klost (exercise sessions)

References:

[CLRS] T. H. Cormen, C. Leiserson, R. Rivest, C. Stein. Introduction to Algorithms, MIT Press 2009

[KT] J. Kleinberg, E. Tardos. Algorithm Design, Addison-Wesley 2005.

[T] R.E. Tarjan. Data Structures and Network Algorithms, SIAM 1987.

Topics:

0. Administrative stuff

1. Introduction

Algorithms, computational models.

RAM model.

Time and space complexity of algorithms.

Worst-case, average-case.

Deterministic and randomized.

Mergesort, Quicksort.

Implementing an initializable array.

Asymptotic analysis.

2. Sorting

Variants of Quicksort.

Analysis of Randomized Quicksort.

Lower bounds.

Information-theoretic lower bound for sorting.

Adversary arguments.

Sorting integers. Counting and Radix Sort.

3. Divide and Conquer

Solving typical recurrences.

Induction, recursion tree, Master theorem.

Karatsuba's algorithm.

4. Selection

Quickselect.

Median of medians.

Optimal randomized selection.

Lower bounds for selection.

5. Dynamic programming

String Edit Distance.

Matrix Chain Multiplication.

General principles.

Optimum binary search trees.

Subset Sum, TSP.

All-pairs shortest paths.

6. Amortized analysis

Binary counter.

Different analyses: direct, bank-account, potential function.

Re-sizable arrays. Variants.

7. Priority queues

Heaps recap, overview.

Skew heaps.

Binomial and Fibonacci heaps.

8. Dictionaries

Binary search trees recap, overview.

Splay trees.

9. Minimum spanning trees

Basics. Generic algorithm.

Concrete algorithms: Boruvka, Jarnik-Prim, Kruskal.

Hybrid algorithm.

10. Maintaining disjoint sets

Iterated functions. Ackermann-hierarchy.

Union-find, simple strategies.

Path compression.

11. Shortest paths

Basics.

SSSP: Dijkstra, Bellman-Ford.

APSP: Floyd-Warshall, Johnson's.

Seidel's algorithm.

12. Network flows

Basics. Overview.

Max flow – min cut. Augmenting paths.

Ford-Fulkerson, Edmonds-Karp.

Blocking flow, Dinitz-algorithm. Special cases.

Applications: bipartite matching, scheduling, vertex- and edge-disjoint paths.

13. Matching

Basics. Augmenting paths.

Edmonds' algorithm.

14. NP-completeness

Motivating examples.

Turing machines, decision problems.

Classes P and NP.

Example problems in NP.

Polynomial reductions.

NP-hard, NP-complete.

SAT, CircuitSAT.

Cook-Levin theorem.

3SAT and various other NP-complete problems.

15. Further complexity-classes

16. Coping with hard problems

Heuristics, special cases.

Approximation: TSP, Vertex Cover.

Exact exponential: TSP, 3SAT, Independent Set.

Parameterized algorithms: Vertex Cover.

17. Linear programming

Basics. Geometric view.

Simplex algorithm.

Integer linear programming.

LP-relaxation.