# Advanced Algorithms (Winter 2023/2024)
## Instructors:
## László Kozma (lectures), Michaela Krüger (exercise sessions)

**Topics:**

**0. Administrative stuff**

**1. Introduction**
Algorithms, computational models.
RAM model.
Time and space complexity of algorithms.
Worst-case, average-case.
Deterministic and randomized.
Asymptotic analysis, O-notation.

**2. Divide and Conquer**
Mergesort, Quicksort.
Partitioning, loop invariants.
Selection problem: good splitter.
Randomized selecton: Quickselect and its analysis.
Deterministic selection: Median-of-medians and its analysis.
Solving typical recurrences.
Direct method, induction, recursion tree.
Master theorem.

**3. Lower bounds**
Information-theoretic lower bounds.
Lower bounds for searching, sorting, merging.
Adversary arguments.
Lower bounds for selection.

**4. Fast multiplication**
Karatsuba’s algorithm.
Fast matrix multiplication (sketched).
Verification of matrix multiplication: Freivalds’ algorithm.
Polynomial evaluation/interpolation.
Fast Fourier Transform.

**5. Dynamic programming**
Example: Interval scheduling.
General principles.
Example: String Edit Distance.
Saving space: Hirschberg’s algorithm.
DP on trees: Weighted Independent Set.
DP on numeric values: Subset Sum.
All-pairs shortest paths: Floyd-Warshall algorithm (sketched).

**6. Amortized analysis**
Binary counter.
Different analyses: bank-account, potential function.
Re-sizable arrays, stacks, and queues. Deamortization.
Extracting small or large elements in O(1) amortized time.

**7. Priority queues**
Heaps recap.
Binomial and Fibonacci heaps and their analysis.

**8. Dictionaries**
Binary search trees recap.
Splay trees (self-adjusting trees) and some properties.
Proof of amortized O(logn) cost of splay trees.
Hashing. Universal families of hash functions.
Hash table with chaining. Perfect hashing (FKS).
Count-min sketch data structure (only idea sketched with no analysis).

**9. Minimum spanning trees**
Basics. Generic algorithm.
Concrete algorithms: Boruvka, Jarnik-Prim, Kruskal.
Hybrid algorithm (exercise)

This part is skipped this year. Optional material in lecture videos, notes:
[Maintaining disjoint sets. Union-find, simple strategies. Path compression.]

**10. Shortest paths**
SSSP: Dijkstra, Bellman-Ford (recap/self-study).
APSP: Floyd-Warshall (dynamic programming – recap).
APSP: Johnson’s algorithm: vertex potentials.
APSP: Seidel’s algorithm: matrix multiplication.

**11. Network flows**
Basics. Overview.
Max flow – min cut. Augmenting paths.
Ford-Fulkerson, Edmonds-Karp two variants: shortest/widest paths.
Optional: Blocking flow, Dinitz-algorithm. Special cases.
Applications: bipartite matching, scheduling, etc.

**12. 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:
Hamiltonian path/cycle, maximum clique, coloring, subset sum.
--- additional material, not for the exam:

Further complexity-classes (omitted)

13. Coping with hard problems
Heuristics, special cases.
Approximation: TSP, Vertex Cover.
Exact exponential: TSP, 3SAT, Independent Set.
Parameterized algorithms: Vertex Cover.

15. Linear programming (optional material provided)
Basics. Geometric view.
Simplex algorithm.
Integer linear programming.
LP-relaxation.

15. Matching theory (not covered this year)
Basics. Augmenting paths.
Edmonds’ algorithm