Algorithm II

2021H

WU Xiaokun 吴晓堃 xkun.wu [at] gmail

Computer Science - UG3/G	
Course	Algorithm II
Term	2021H
Final	Tba
Credits	2
Staff	WU Xiaokun 吴晓堃
Lecture	32 hours

Course Information

Short Intro

The objective of this course is to provide a complete introduction to *algorithm design* and *complexity analysis* techniques.

Description

The course aims at providing fundamental knowledge and existing techniques of algorithm design-related topics. It also aims at elaborating rigorous complexity analysis for a better understanding of the design principles. The course should follow a programming-focused introductory computer science sequence.

The course will touch on the following topics:

- Basic concepts recap: tractability, asymptotic order, graphs.
- Major algorithm design techniques: greedy algorithms, divide and conquer, dynamic programming, network flow.
- Computational intractability: $\mathcal{NP}$, NP-Complete, PSPACE.
- Dealing with intractable problems: identification of structured special cases, approximation algorithms, local search heuristics.
- Randomized algorithms.

Keywords: algorithm design, complexity analysis.

Prerequisites

Required:

- Sufficient programming experience.
- Comfortable with mathematical proofs.

Recommended:

- Knowledge in computer science fundamentals: data structure, operating system, computer architecture, etc.
- As much knowledge of mathematics as possible.
- Insights in your own specific area of study.

Teaching plan

The course is organized into several major parts and each contains different topics.

Introduction & basic concepts recap:

- Computational Tractability
- Asymptotic Order of Growth
- Graphs

Major algorithm design techniques:

- Greedy algorithms,
- Divide and conquer,
- Dynamic programming,
- Network flow.

$\mathcal{NP}$ and PSPACE:

- Reducibility

- $\mathcal{P} \neq \mathcal{NP}$?
- NP-Complete
- $\mathcal{P} \neq$ PSPACE?
- PSPACE-complete

Dealing with intractability:

- Identification of structured special cases,
- Approximation algorithms,
- Local search heuristics,
- Randomized algorithms.

Schedule

Friday S6-S9, Library 916.

Week	Date	Lecture	Handouts
1	2021/10/15	Introduction	Stable Matching
2	2021/	Algorithm Analysis & Graphs	Tractability & Asymptotic Order, Graphs
3	2021/	Greedy Algorithm I	Coin Charging, Interval Scheduling/Partitioning, Minimize Lateness, Optimal Cache
4	2021/	Greedy Algorithm II	Dijkstra's Algorithm, Minimum Spanning Tree, Prim/Kruskal/Boruvka, Clustering, Min-cost Arborescence, Huffman Codes
5	2021/	Divide and Conquer I	Mergesort, Recurrence Relations, Counting Inversions, Randomized quicksort, Closest Pair of Points
6	2021/	Divide and Conquer II	Master Theorem, Integer/Matrix Multiplication, Convolutions & FFT
7	2021/	Dynamic Programming I	Weighted Interval Scheduling, Segmented Least Squares, Subset Sums and Knapsacks
8	2021/	Dynamic Programming II	Sequence Alignment, Hirschberg/Bellman–Ford–Moore, Negative Cycle Detection
9	2021/	Network Flow I	Max-Flow Min-Cut, Ford-Fulkerson, Augmenting Paths, Capacity-scaling/Shortest-augmenting/Dinitz, Simple unit-capacity Networks
10	2021/	Network Flow II	Bipartite Matching, Disjoint Paths, Circulations with Demands, Survey Design, Airline Scheduling, Image Segmentation, Project Selection, Tournament Elimination
11	2021/	Intractability I	Polynomial-Time Reductions, Packing & Covering problem, Satisfiability Problem (raster/SAT), Traveling Salesman, Partitioning Problems, Graph Coloring, Numerical Problems
12	2021/	Intractability II	$\mathcal{P}$ vs. $\mathcal{NP}$, $\mathcal{NP}$-complete, co-$\mathcal{NP}$, $\mathcal{NP}$-hard
13	2021/	Extending Tractability	Special cases (tree, planarity), Approximation (vertex cover, knapsack), Exponential Algorithms (3-SAT, TSP)
14	2021/	PSPACE	Quantified 3-SAT (QSAT), Planning Problems, PSPACE-complete
15	2021/	Approximation Algorithms	Load Balancing, Center Selection, Weighted Vertex Cover, Knapsack Problem
16	2021/	Local Search	Gradient Descent, Metropolis & Simulated Algorithm, Hopfield Neural Networks, Maximum-Cut, Nash Equilibria
S1	2021/	Randomized Algorithms	Contention Resolution, Global Min-Cut, Max 3-Satisfiability, Universal Hashing, Chernoff Bounds

Note: slides content are largely consistent with the official accompanying slides of *Algorithm Design*.

Evaluation

- Attendance & participation: 10%
- Understanding of the topic: 50%

- Presenting skill: 20%
- Final project: 20%
- Honorable bonus: 10%

Textbook

Not mandatory but recommended:

- Kleinberg & Tardos, *Algorithm Design*.
- Sedgewick & Wayne, *Algorithms*.
- Cormen et al., *Introduction to Algorithms*.

Resources

- It's possible to find PDF files from the web for all textbooks listed above.
- [Lecture Slides for Algorithm Design][1]
- [Course site][2]

---

1. https://www.cs.princeton.edu/~wayne/kleinberg-tardos/ ↩
2. https://xlanwu.github.io/teach/Algorithm/2021H.html ↩