KAU CS 611 • Tentative (2022–23)

Advanced Analysis of Algorithms

Dr. Muhammad Al-Hashimi

Credits 3

Prerequisite None • See *Undergrad Background*

Homepage www.hashimi.ws/cs611

Email cs611@hashimi.ws

Office Room 139 (B31) ✆ Ext. TBA

Graduate course on the analysis, design, and complexity of computer-based solutions. It aims to cover advanced topics with sound theory and to equip students with the needed skills to pursue advanced work. We introduce students to the main tools and methods used to analyze algorithms. They explore algorithmic design, parallel techniques, NP-completeness, and other course topics. Case studies, short investigations, and exposure to related seminal works and sources guide their learning. The course includes a programming component to help develop an experience-based understanding of core issues. Check the official catalog description for details.

Topics

- ✎ **Analysis Foundation** Formalizing efficiency, nonrecursive and recursive iteration, mathematical induction, sequences, sums, recurrences and generating functions, Master theorem, binary heaps and integer priority queues.
- ✎ **Parallel Fundamentals** PRAM and other parallel models, Brent's principle, parallel priority queues, divide-conquer and other basic techniques.
- ✎ **Intractability** Decision problems, P and NP, polynomial transformations, NP-completeness, Cook's Theorem, 3-staisfiability and the basic 21 NP-Complete, branch-bound/backtracking.
- ✎ **Case Study Algorithms** Quickselect, mergesort, quicksort, polynomial evaluation and the fast Fourier transform, Prim's minimum spanning tree/Dijkstra's single-source-shortest-path (SSSP), parallel SSSP, the knapsack problem.

Resources Check announcement conversations in course group for schedules, and material.

Undergrad Background Anany Levitin, *Introduction to the Design and Analysis of Algorithms*, Pearson 3rd edition.

Assessment Short case-study investigations, presentations, and group discussions assess work quality and grasp of course material.

20% Homework exercises
`(CLO/Synth) Develop necessary skills with standard methods and tools used to analyze algorithms`

40% Projects
`(CLO/Anal) Examine various characteristics of interest to the design and performance of algorithms`

40% Final exam/presentation
`(CLO/Appl) Demonstrate a satisfactory grasp of concepts, techniques, and main results from the complexity of algorithms`

References

1. Cormen, Leiserson, Rivest, and Stein, *Introduction to Algorithms*, The MIT Press 4th edition (2022).
2. Peter Sanders, Kurt Mehlhorn, Martin Deitzfelbinger, and Roman Dementiev, *Sequential and Parallel Algorithms and Data Structures: The Basic Toolbox*, Springer 2019.
3. Garey and Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman 1979 (1983 printing or later).
4. Velleman, *How to Prove It*, Cambridge University Press 3rd edition (2019).
5. Graham, Knuth, and Patashnik, *Concrete Mathematics: A Foundation for Computer Science*, Addison-Wesley 1st or 2nd editions (a minor difference).
6. Bentley, Haken, and Saxe, *A General Method for Solving Divide-and- Conquer Recurrences*, SIGACT News (1980). **DOI** 10.1145/1008861.1008865
7. Sedgewick 2022 (talk, Cambridge UK), *What do we know about Quicksort?*, author website (`sedgewick.io`).
8. Sedgewick 1977, *The Analysis of Quicksort Programs*, Acta Informatica. **DOI** 10.1007/BF00289467.
9. Hoare 1962, *Quicksort*, The Computer Journal. **DOI** 10.1093/comjnl/5.1.10
10. Karp 1972, *Reducibility among Combinatorial Problems*, Complexity of Computer Computations (The IBM Research Symposia Series). **DOI** 10.1007/978-1-4684-2001-2_9

Rev 1.0 (Sep 1, 2022)