Design and Analysis of Algorithms

Fall 2022

Course Description

- This course presents the fundamental techniques to **design efficient algorithms** and **analyze their running time**

Course Description

- After a brief review of prerequisite material (search, sorting, asymptotic notation), we will discuss efficient algorithms for basic graph problems and solving various problems through divide and conquer algorithms, dynamic programming and greedy algorithms

Course Description

- We will consider randomized algorithms, proofs of NP-completeness, approximation algorithms, partial recursive functions, and proofs of undecidability

Course Description

- **Review of algorithm analysis:** Search in ordered array, binary insertion sort, merge sort, worst-case and average-case time complexity, minimum complexity of sorting n elements for small n, 2-3 trees, asymptotic notation (O, Ω, Θ, o, ω)
- **Divide and conquer algorithms:** Master theorem, integer multiplication, matrix multiplication, fast Fourier transform

Course Description

- **Graphs:** Breadth-first search, connected components, topological ordering, depth-first search, from planar graphs to Robertson-Seymour theorem
- **Dynamic programming:** Chain matrix multiplication, shortest paths, edit distance, sequence alignment, extensions of dynamic programming
- **Greedy algorithms:** Binary heaps, Dijkstra's algorithm, minimum spanning tree, Huffman codes, matroids

Course Description

- **Randomized algorithms:** Selection, quick sort, global minimum cut, hushing
- **P and NP:** Cook's theorem, examples of NP-complete problems
- **Approximate algorithms for NP-hard problems and polynomial algorithms for subproblems of NP-hard problems:** Set cover, vertex cover, maximum independent set, 2-SAT

Course Description

- **Partial recursive functions:** Theorem of Post, Diophantine equations
- **Computations and undecidable problems:** Existence of complex problems, undecidability of halting problem, theorem of Rice, semantic and syntactical properties of programs

Time and Place

- Monday, Thursday, 14:30-16:00
- Building 9, Room 4223

Possible Reading

All required information is in presentations

- Algorithm Design, by J. Kleinberg and E. Tardos, Addison-Wesley, 2005
- Introduction to Algorithms (3rd Edition), by T. Cormen, C. Leiserson, R. Rivest, and C. Stein, The MIT Press, 2009

Possible Reading

- Algorithms, by S. Dasgupta, C. Papadimitriou, and U. Vazirani, McGraw-Hill, 2006
- Theory of Recursive Functions and Effective Computability, by H. Rogers, McGraw-Hill, 1967
- Computers and Intractability. A Guide to the Theory of NP-Completeness, by M.R. Garey and D.S. Johnson, W.H. Freeman and Company, 1979
- Introduction to Algorithm Complexity, by V. Alekseev, Moscow State University, 2002 (in Russian)

Prerequisites

- Computer programming skills
- Knowledge of probability
- Understanding of basic data structures and algorithms
- Basic knowledge in discrete mathematics

Grading

- Course work will consist of homework assignments, two midterm exams, and project
- Tentative weights:
  - homework 40%,
  - midterm exams 30%,
  - project 30%,
- Note that these weights are subject to change
- Students should work with homework assignments in groups (usually, 3-5 students in a group)

Contact Details

- **Instructor: Mikhail Moshkov**
- mikhail.moshkov@kaust.edu.sa

- Office Hours: Tuesday 15:30-17:00
- Building 1, Room 4108 (level four, sea side, the leftmost room)

Contact Details

- **Teaching Assistant:** Zainab Alsuwaykit
- zainab.alsuwaykit@kaust.edu.sa
- Office Hours: TBD
- Building TBD

Project

- Groups of 3-5 students (at least two with relatively good programming skills)

Project

- It is necessary
  - to chose a problem
  - to choose two different algorithms for this problem solving
  - to find theoretical results about time complexity of these algorithms
  - to create software
  - to make experiments
  - to compare theoretical and experimental results
  - to write report

Project

- The first four weeks:
  - organization of groups, choice of problems and algorithms
  - preparation of proposal for each group containing title, list of members, descriptions of problem and two algorithms, and references
- Beginning of the fifth week – submission of proposal
- Beginning of the sixth week – approval of proposal

Project

Grading:

- proposal 6%
- midterm presentation 6%
- midterm report 6%
- final presentation 6%
- final report 6%

Exams

- Midterm Exam 1, Oct 20
- Midterm Exam 2, Nov 28

Tentative Schedule

- 1 Mon 08/29/2022 Search and Sorting
- 1 Thu 09/01/2022 Search and Sorting
- 2 Mon 09/05/2022 Search and Sorting
- 2 Thu 09/08/2022 Divide and Conquer Algorithms
- 3 Mon 09/12/2022 Divide and Conquer Algorithms
- 3 Thu 09/15/2022 Divide and Conquer Algorithms
- 4 Mon 09/19/2022 Graphs
- 4 Thu 09/22/2022 Saudi National Day

Tentative Schedule

- 5 Mon 09/26/2022 Graphs, Project Proposal
- 5 Thu 09/29/2022 Graphs
- 6 Mon 10/03/2022 Dynamic Programming
- 6 Thu 10/06/2022 Dynamic Programming
- 7 Mon 10/10/2022 Dynamic Programming, HW1
- 7 Thu 10/13/2022 Greedy Algorithms
- 8 Mon 10/17/2022 Mid-Semester Break
- 8 Thu 10/20/2022 Midterm Exam 1

Tentative Schedule

- 9 Mon 10/24/2022 Midterm Presentation of Project
- 9 Thu 10/27/2022 Greedy Algorithms, Midterm Project Report
- 10 Mon 10/31/2022 Randomized Algorithms
- 10 Thu 11/03/2022 Randomized Algorithms
- 11 Mon 11/07/2022 P and NP
- 11 Thu 11/10/2022 P and NP
- 12 Mon 11/14/2022 Work with NP-Hard Problems
- 12 Thu 11/17/2022 Work with NP-Hard Problems

Tentative Schedule

- 13 Mon 11/21/2022 Partial Recursive Functions, HW2
- 13 Thu 11/24/2022 Partial Recursive Functions
- 14 Mon 11/28/2022 Midterm Exam 2
- 14 Thu 12/01/2022 Computations and Unsolvable Problems
- 15 Mon 12/05/2022 Computations and Unsolvable Problems, Final Presentation of Project, HW3
- 15 Thu 12/08/2022 Final Presentation of Project, Final Project Report