جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

Course Syllabus: Design and Analysis of Algorithms - CS 260

Division	Computer, Electrical and Mathematical Sciences & Engineering
Course Number	CS 260
Course Title	Design and Analysis of Algorithms
Academic Semester	Spring
Academic Year	2017/2018
Semester Start Date	01/28/2018
Semester End Date	05/24/2018
Class Schedule (Days & Time)	01:00 PM - 02:30 PM | Mon Thu

Instructor(s)

Name	Email	Phone	Office Location	Office Hours
Mikhail Moshkov	mikhail.moshkov@kaust.edu.sa	+966128080334		Wednesday, 3:30 - 5:00 pm

Teaching Assistant(s)

Name	Email
Michal Mankowski	michal.mankowski@kaust.edu.sa

Course Information

Comprehensive Course Description	The course covers main approaches to design and analysis of algorithms including important algorithms and data structures, and results in complexity and computability. The main contents are: 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); divide and conquer algorithms (master theorem, integer multiplication, matrix multiplication, fast Fourier transform); graphs (breadth-first search, connected components, topological ordering, depth-first search, way 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); 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 or polynomial algorithms for subproblems of NP-hard problems (set cover, vertex cover, maximum independent set, 2-SAT); 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).
Course Description from Program Guide	The course covers main approaches to design and analysis of algorithms including important algorithms and data structures, and results in complexity and computability. The main contents are: 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); divide and conquer algorithms (master theorem, integer multiplication, matrix multiplication, fast Fourier transform); graphs (breadth-first search, connected components, topological ordering, depth-first search, way 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, Dijkstras algorithm, minimum spanning tree, Huffman codes, matroids); randomized algorithms (selection, quick sort, global minimum cut, hushing); P and NP (Cooks theorem, examples of NP-complete problems); approximate algorithms for NP- hard problems or polynomial algorithms for subproblems of NP-hard problems (set cover, vertex cover, maximum independent set, 2-SAT); 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).

Goals and Objectives	The main goal of this course is to study the fundamental techniques to design efficient algorithms and analyze their running time. 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. We will consider also randomized algorithms, proofs of NP-completeness, approximation algorithms, partial recursive functions, and proofs of undecidability.
Required Knowledge	1. Computer programming skills 2. Knowledge of probability 3. Understanding of basic data structures and algorithms 4. Basic knowledge in discrete mathematics
Reference Texts	1. Algorithm Design, by J. Kleinberg and E. Tardos, Addison-Wesley, 2005 (main textbook) 2. Introduction to Algorithms (3rd Edition), by T. Cormen, C. Leiserson, R. Rivest, and C. Stein, The MIT Press, 2009 3. Algorithms, by S. Dasgupta, C. Papadimitriou, and U. Vazirani, McGraw-Hill, 2006 4. Theory of Recursive Functions and Effective Computability, by H. Rogers, McGraw-Hill, 1967 5. 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 6. Introduction to Algorithm Complexity, by V. Alekseev, Moscow State University, 2002 (in Russian) All required for the course information is in presentations
Method of evaluation	**30.00**% - Homework /Assignments **20.00**% - Research Project **20.00**% - Midterm exam **30.00**% - Final exam
Nature of the assignments	Course work will consist of homework assignments, midterm exam, project, and final comprehensive exam. In the 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 prepare proposal, to make two presentations, and to write two reports. For project: proposal 4%, midterm presentation 4%, midterm report 4%, final presentation 4%, final report 4%
Course Policies	Students should work with homework assignments and with projects in groups (usually, 3-4 students in a group)
Additional Information	

Tentative Course Schedule *(Time, topic/emphasis & resources)*		
Week	**Lectures**	**Topic**
1	Mon 01/29/2018 Thu 02/01/2018	Search and Sorting
2	Mon 02/05/2018 Thu 02/08/2018	Search and Sorting
3	Mon 02/12/2018 Thu 02/15/2018	Divide and Conquer Algorithms
4	Mon 02/19/2018 Thu 02/22/2018	Graphs, Project Proposal
5	Mon 02/26/2018 Thu 03/01/2018	Graphs, HW1
6	Mon 03/05/2018 Thu 03/08/2018	Dynamic Programming
7	Mon 03/12/2018 Thu 03/15/2018	Dynamic Programming
8	Mon 03/19/2018 Thu 03/22/2018	Greedy Algorithms, Midterm Presentation of Project
9	Mon 03/26/2018 Thu 03/29/2018	Randomized Algorithms, Midterm Project Report, HW2
10	Mon 04/02/2018 Thu 04/05/2018	Spring Break
11	Mon 04/09/2018 Thu 04/12/2018	P and NP
12	Mon 04/16/2018 Thu 04/19/2018	Work with NP-Hard Problems, Midterm Exam
13	Mon 04/23/2018 Thu 04/26/2018	Work with NP-Hard Problems
14	Mon 04/30/2018 Thu 05/03/2018	Partial Recursive Functions
15	Mon 05/07/2018 Thu 05/10/2018	Computations and Unsolvable Problems
16	Mon 05/14/2018 Thu 05/17/2018	Computations and Unsolvable Problems, Final Presentation of Project, Final Project Report, HW3
17	Mon 05/21/2018 Thu 05/24/2018	Final Comprehensive Exam
18		

Note

The instructor reserves the right to make changes to this syllabus as necessary.