Imdad Ullah Khan

Associate Professor
Department of Computer Science
Syed Babar Ali School of Science and Engineering
Lahore University of Management Sciences (LUMS), Pakistan
Email: imdad.khan@lums.edu.pk

Home Research Teaching Services Students

CS510 - Design and Analysis of Algorithms

Course Description:

This is a graduate-level course on the design and analysis of algorithms. After a quick review of basic algorithm design techniques such as divide-and-conquer, greedy algorithms, and dynamic programming, the course covers introductory complexity theory and advanced topics in algorithms such as network flows, randomized algorithms, approximation algorithms, and linear programming.

Prerequisites:

An undergraduate course in Discrete Mathematics, Data Structures, Algorithms and Probability.
Strong problem solving skills and some background in programming.

Textbook(s)/Supplementary Readings:

Algorithm Design
Jon Kleinberg and Éva Tardos

Lectures:

Topic	Slides	Notes	Problem Sets
Algorithmic Thinking	Slides	Notes	
Problem Formulation Multiplication example using pen and paper			
Asymptotic Analysis		Notes	PS
Runtime Analysis ang Big Oh	Slides		
Complexity Classes and Curse of Exponential Time	Slides		
Relational Properties: Big Omega, Big Theta, Little Oh	Slides		

Topic	Slides	Notes	Problem Sets
Design Paradigm: Divide and Conquer		Notes	PS
Quick Sort	Slides		
Karatsuba Algorithm for Integers Multiplication	Slides		
Finding Closest Pair in Plane	Slides		
Convex Hull			
Single Source Shortest Path		Notes	PS
Weighted Graphs and Shortest Paths	Slides		
Dijkstra Algorithm	Slides		
Proof of Correctness	Slides		
Implementation & Runtime	Slides		
Prim's Algorithm		Notes	PS
Minimum Spanning Tree	Slides		
Prim's Algorithm for MST	Slides		
Cuts in Graphs	Slides		
Correctness and Optimality	Slides		
Implementation & Runtime	Slides		
Kruskal's Algorithm		Notes	
The Cycle Property (Red Rule)	Slides		
Reverse Delete Algorithm for MST			
Kruskal's Algorithm for MST	Slides		
Runtime and Implementation	Slides		
Disjoint Sets Data Structure			
Dynamic Programming	Slides	Notes	PS
Computing Fibonacci Numbers			
Optimal Substructure			
Memoization			
Weighted Independent Sets			
(Weighted) Independent Set in Graphs	Slides		
Weighted Independent Sets in Path	Slides		
Dynamic Programming Formulation	Slides		
Implementation and Backtracking	Slides		
The Knapsack Problem			
The Knapsack Problem	Slides		
Dynamic Programming Formulation	Slides		
Implementation	Slides		
Fractional Knapsack and Subset Sum Problem	Slides		
Sequence Alignment			
Sequence Analysis	Slides		
The Sequence Alignment Problem	Slides		
Dynamic Programming Formulation	Slides		
Matrix Chain Multiplication			

Topic	Slides	Notes	Problem Sets
Matrix Chain Multiplication			
Intractable Problems			
Efficiently Solvable vs. Hard (Intractable) Problems	Slides		
Clique, Independent Set	Slides		
Vertex Cover, Set Cover, Set Packing	Slides		
Hamiltonian Cycle, Path and TSP	Slides		
Coloring and Partitioning	Slides		
Numeric and Number Theoretic Problems	Slides		
Satisfiability Problems	Slides		
Polynomial Time Reduction		Notes	PS
Polynomial Time Reduction Definition	Slides		
Reduction by Equivalence	Slides		
Reduction from Special Cases to General Case	Slides		
Reduction by Encoding with Gadgets-I	Slides		
Reduction by Encoding with Gadgets-II	Slides		
Transitivity of Reductions	Slides		
Decision, Search and Optimization Problem	Slides		
Self-Reducibility	Slides		
Classes of Problems			
Polynomial Time Verification	Slides		
The Classes P and NP	Slides		
The Classes EXP and coNP	Slides		
NP-Hard and NP-Complete Problems	Slides		
Proving NP-Hardness	Slides		
A First NP-Complete Problem	Slides		
Proving NP-Complete Problems	Slides		
The Cook-Levin Theorem: SAT is NP-Complete			
NP-Complete Problems from known Reductions			
NP-Complete Applications			
Coping with NP-Hardness		Notes	
Strategies to deal with hard problems	Slides		
Algorithms for Special Cases	Slides		
Fixed Parameter Tractability	Slides		
Intelligent Exhaustive Search			
Backtracking	Slides		
Branch and Bound	Slides		
Dynamic Programming for TSP	Slides		
Approximation Algorithms		Notes	PS
Approximation Algorithms for Optimization Problems	Slides		
Absolute Approximation Algorithms	Slides		
Inapproximability by Absolute Approximate Algorithms	Slides		
Relative Approximation Algorithm	Slides		
Inapproximability by Relative Approximate Algorithms	Slides		
Polynomial Time Approximation Schemes			

Topic	Slides	Notes	Problem Sets
Fully Polynomial Time Approximation Schemes	Slides		
	Slides		
Randomized Algorithms		Notes	PS
Deterministic and Randomized Algorithms	Slides		
Probability Review	Slides		
Probabilistic Analysis of Quick-Sort Algorithm	Slides		
Randomized-Select	Slides		
Max-Cut	Slides		
Min-Cut	Slides		
Max-3-Sat and Derandomization	Slides		
Closest Pair	Slides		
Hashing, Bloom Filters, Streams	Slides		
Local Search	Slides	Notes	
Local Search for MAX-CUT			
Gradient Descent			
Metropolis Algorithm			
Simulated Annealing			

Homeworks

In this homework, we will develop our concepts for divide and conquer based algorithms for sorting an array of n numbers. [Homework]
With this homework, we will develop a divide and conquer based algorithm for the general problem of selection of an order statistics of an array. [Homework]
Homework-3 is specially designed for thoroughly understanding the concepts of Bipartite-Graph and matching. We will develop a combinatorial algorithm for matching in Bipartite-Graphs. This is a classic combinatorial optimization problem with many applications. The algorithm is very simple and simply beautiful. [Homework]
With this set of problems, we will develop algorithms for All-Pairs Shortest Paths (APSP) in graphs. [Homework]

Back to Top