Advanced Algorithms (CS6L007)

Books
1. **[DPV]** Algorithms - Sanjay Dasgupta, Christos Papadimitriou, Umesh Vazirani
2. **[KT]** Algorithm Design - Jon Kleinberg and Eva Tardos
3. **[CLRS]** Introduction to Algorithms (Third Edition) - Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein

----

Week 1: Lectures (July 25, 2024) and (July 26, 2024):
a) Introduction to Algorithms, Correctness and Efficiency of an algorithm, case study: Fibonacci Numbers and Insertion Sort.
b) Worst-case analysis of an algorithm
c) Models of computation - Word-RAM model
d) Asymptotic Analysis: Big O, Omega, Theta
e) Some examples of different upper bounds

Reference: Chapter 1 in DPV, Chapter 2 in KT, Chapter 2 (Section 2.1 and 2.2) in CLRS

Week 2: Lectures (August 1, 2024) and (August 2, 2024):
Algorithm Design Technique 1: Divide and Conquer
a) Multiplying two n-bit numbers (Karatsuba and Ofman)
b) Proof of Master Theorem
c) Matrix Multiplication (Strassen)
d) A problem in computational geometry : Finding closest pair of points
e) Counting Inversions
f) Finding Maximum Sum Subarray

Reference: Chapter 2 in DPV, Chapter 4 in KT, Section 4.1 in CLRS

Week 3: Lectures (August 8, 2024) and (August 9, 2024):

Proving Lower Bounds

a) Lower bound using Decision Tree model (Information Theoretic Argument) : Case study: Comparison Sorting Algorithms, Searching a sorted sequence, Merging two sorted sequence
b) Limitation of Information Theoretic Argument and need for Adversary Argument
c) Lower bound using Reduction

Reference: Toniann Pitassi's **Notes** and Section 8.1 in CLRS

Week 4: Lectures (August 15, 2024: Holiday) and (August 16, 2024):

Graph Algorithms

a) s-t connectivity problem
b) Correctness proof of Breadth-First Search (BFS), BFS tree properties
c) Application of BFS: Testing Bipartiteness and Finding diameter in a tree
d) Depth-First Search (DFS) in directed graphs and DFS tree properties
e) Application of DFS in directed graphs: Detecting Directed Acyclic Graphs, Topological Sort, Finding Strongly Connected Components

Reference: Chapter 3 in KT, Chapter 22 in CLRS

Week 5: Lectures (August 22, 2024) and (August 23, 2024):

Amortized Analysis

a) Aggregate Method
b) Accounting Method
c) Potential Method
d) Dynamic Tables (Expansion and Contraction)

Reference: Chapter 17 in CLRS

Week 6: Lectures (August 29, 2024) and (August 30, 2024):

Algorithm Design Technique 2: Greedy

a) Interval Scheduling Problem: optimality proof using “ Greedy stays ahead” strategy
b) Interval Coloring Problem: optimality proof using structural lower bound argument
c) Interval Scheduling to minimize lateness: optimality proof using exchange argument
d) Introduction to Matroid Theory

Reference: Chapter 5 in KT and Section 16.4 in CLRS

Week 7: Lectures (September 5, 2024) and (September 6, 2024 : Revision slot):

Matroid Theory

a) Graphic Matroid for an Undirected Graph
b) Solving Minimum Spanning Tree problem by designing a greedy algorithm for a Weighted Graphic Matroid
c) Proof that the greedy algorithm for the Graphic Matroid returns an optimal solution and an optimal solution for the graphic matroid is an optimal solution for the minimum spanning tree problem
d) A quick recap of Kruskal and Prim’s algorithms - Cut-Property for proof of correctness

Reference: Chapter 5 in KT and Section 16.4 in CLRS

Week 8: Mid-Semester Exam Week

Week 9: Lectures (September 19, 2024 : Break post exam) and (September 20, 2024):

Algorithm Design Technique 3: Dynamic Programming

a) Weighted Interval Scheduling Problem :
   Top-Down approach: Recursion and Smart Recursion (Memoization)
   Bottom-Up approach: Iterative method
b) Principles of Dynamic Programming design
c) Longest Increasing Subsequence Problem

Reference: Sections 6.1, 6.2 in KT and Section 6.2 in DPV

Week 10: Lectures (September 26, 2024) and (September 27, 2024):

More on Dynamic Programming

a) Subset Sums Problem
b) Knapsack Problem: With repetition and Without repetition
c) Rod Cutting Problem
d) Edit Distance Problem
e) Symmetric Traveling Salesman Problem
f) Maximum Independent Set in Trees

Reference: Sections 6.4 in KT, Sections 6.3, 6.4, 6.6, 6.7 in DPV, and 15.1 in CLRS

Week 11: Lectures (October 3, 2024) and (October 4, 2024):

Algorithm Design Technique 4: Network Flow Algorithms

a) The Maximum Flow Problem, Residual Graphs, Augmenting Paths
b) Ford-Fulkerson algorithm, Running time analysis
c) Correctness of Ford-Fulkerson algorithm: Max-Flow Min-Cut theorem
d) Application 1 of Max Flow algorithms (Maximum Bipartite Matching)

Reference: Sections 7.1, 7.2, and 7.5 in KT

Week 12: Lectures (October 10, 2024) and (October 11, 2024):

Institute Holidays : No Lectures

Week 13: Lectures (October 17, 2024) and (October 18, 2024):

More on Network Flows:

a) Scaling Max Flow algorithm
b) Dinitz Edmond and Karp algorithm
c) Application 2 of Max Flow algorithms (Maximum Edge Disjoint Paths)

Reference: Sections 7.3 and 7.6 in KT, Theorem 26.8 in CLRS

Week 14: Lectures (October 25, 2024) and (October 26, 2024):

NP-completeness

a) Polynomial time reductions
    - Independent Set to Vertex Cover
    - Vertex Cover to Independent Set
    - Vertex Cover to Set Cover
    - Independent Set to Set Packing
    - 3-SAT to Independent Set
b) The class P and NP
c) NP-Complete problems

Reference: Sections 8.1, 8.2, 8.3 , and 8.4 in KT

Week 15: Lectures (October 31, 2024) and (November 1, 2024):
No class: Diwali Holidays

Week 16: Lectures (November 7, 2024) and (November 8, 2024):
More on NP-completeness
a) The first NP-complete problem - circuit satisfiability
b) Circuit Satisfiability to SAT, SAT to 3-SAT
c) General Strategy to prove a problem NP-complete
d) 3-coloring is NP-complete using 3-SAT

Approximation Algorithms: How to cope with NP-completeness
a) 2-approximation algorithm for vertex cover
b) 2-approximation algorithm for Traveling Salesman problem with Triangle Inequality
c) (ln n + O(1)) approximation for Set Cover problem
d) 4 \sqrt{n} approximation for coloring 3-colorable graphs

Reference: Sections 8.4 and 8.7 in KT, Sections 35.1, 35.2, 35.3 in CLRS, Section 6.5 from the book **The Design of Approximation Algorithms by David P. Williams and David B. Shmoys** (the book available online for free).

Week 17: Lectures (November 14, 2024)

More on Approximation Algorithms:
a) If P is not equal to NP, then for any constant c >=1, there is no polynomial time approximation algorithm with approximation ratio c for general Traveling Salesman Problem.
b) 2-approximation algorithm for k-center problem
c) If P is not equal to NP, then for any constant 1 <= c < 2, there is no polynomial time approximation algorithm with approximation ratio c for k-center problem.

Reference: Sections 35.2 in CLRS and Section 2.2 from the book **The Design of Approximation Algorithms by David P. Williams and David B. Shmoys** (the book available online for free).