REVIEW
for
MIDTERM EXAM

Chapters

- Introduction (001Introduction.pdf)
- Ch. 1. Introduction: Representative Problems (01StableMatching.pdf)
  - 01DemoGaleShapley.pdf
- Ch. 2. Algorithm Analysis (02AlgorithmAnalysis.pdf)
- Ch. 3. Algorithm Analysis (03Graphs.pdf)
  - 03demo-dag.pdf
- Ch. 4. Greedy Algorithms (04greedy.pdf, 04mst.pdf, 04huffman.pdf)
  - 04demo-dijkstra.pdf
  - 04demo-mst
- Ch. 5. Divide and Conquer (05divide –and-conquer.pdf, 05multiply.pdf)
  - *05demo-merge.pdf*
  - *05demo-merge-invert.pdf*

Introduction
(001Introduction.pdf)

- Algorithms: What, *Why*, How?
- Algorithm Etymology
- Proof of Correctness

Ch.1. Introduction:Representative Problems

(01StableMatching.pdf) (01DemoGaleShapley.pdf)

- 1.1. Stable Matching
  - Stable matching / Stable marriage problem
  - Perfect matching
  - Unstable pair
  - Gale-Shapley Algorithm
    - Proof of correctness: Termination
    - Proof of correctness: Perfection
    - Proof of correctness: Stability
    - Efficient Implementation
    - Understanding the Solution
      - Valid Partner
      - Man Optimality
      - Women Pessimality

Ch.1. Introduction:Representative Problems

(01StableMatching.pdf) (01DemoGaleShapley.pdf)

- 1.2. Five Representative Problems
  - Interval scheduling
  - Weighted interval scheduling
  - Bipartite matching
  - Independent set
  - Competitive facility location

Ch.2. Algorithm Analysis
(02AlgorithmAnalysis.pdf)

- 2.1. Computational Tractability
  - Motivation
  - Brute-force
  - Why complexity matters?
  - Worst-case analysis
  - Polynomial running time
- *2.2. Asymptotic Order of Growth*
  - Formalization
    - Upper bound (Big Oh) (O)
    - Lower bound (Big Omega) (Ω)
    - Tight bound (Big Theta) (Θ)
    - Properties
  - Asymptotic Bounds for Some Common Functions
- *2.4. Survey of Common Running Times*

- Linear time: $O(n)$ - Linearithmic time: $O(n \log n)$
- Quadratic time: $O(n^2)$ - Cubic time: $O(n^3)$
- Polynomial time: $O(n^k)$ - Exponential time $O(r^n)$

Ch.3. Algorithm Analysis

(03Graphs.pdf) (03demo-dag.pdf)

- 3.1. Basic Definitions and Applications
  - Undirected graphs
  - Graph Representation: Adjacency Matrix
  - Graph Representation: Adjacency List
  - Paths and Connectivity
  - Cycles
  - Trees
  - Rooted Trees
  - Phylogeny Trees
- 3.2. Graph Traversal
  - Connectivity
  - Breadth First Search
    - Algorithm
    - Analysis
  - Depth First Search
  - Connected Component

Ch.3. Algorithm Analysis

(03Graphs.pdf) (03demo-dag.pdf)

- 3.4. Testing Bipartiteness
  - Bipartite Graphs
  - Testing Bipartiteness
    - An Obstruction to Bipartiteness: Odd-length cycle
  - Bipartite Graphs
- 3.5. Connectivity in Directed Graphs
  - Directed Graphs
  - Graph Search
  - Strong Connectivity
    - Mutually reachable
    - Algorithm (G and Grev)
- 3.6. Directed Acyclic Graphs and Topological Ordering
  - Directed Acyclic Graphs
  - Topological order
  - Precedence Constraints
  - *Algorithm & Running Time*

Ch.4. Greedy Algorithms
(04greedy.pdf)

- 4.1. Interval Scheduling / Interval Partitioning
  - Interval Scheduling: Greedy Algorithms
    - Earliest start time
    - Shortest interval
    - Fewest conflicts
    - Earliest finish time
      - Analysis
  - Interval Partitioning
    - Lower Bound on Optimal Solution
    - Greedy Algorithm
    - Greedy Analysis

Ch.4. Greedy Algorithms
(04greedy.pdf)

- 4.2. Scheduling to Minimize Lateness
  - Minimizing lateness problem
  - Minimizing Lateness: Greedy Algorithms
    - Shortest processing time first
    - Smallest slack
    - Earliest deadline first
    - Minimizing Lateness:
      - Greedy Algorithm
      - No Idle Time
      - Inversions
      - Analysis of Greedy Algorithm

Ch.4. Greedy Algorithms
(04greedy.pdf)

- 4.3. Optimal Caching
  - Optimal Offline Caching
    - Caching
    - Cash hit
    - Cash miss
    - Farthest-In-Future
  - Reduced Eviction Schedules
  - Multiple Optimal Schedules
  - Farthest-In-Future: Analysis

Ch.4. Greedy Algorithms

(04greedy.pdf) (04demo-dijkstra.pdf)

- 4.4. Shortest Paths in a Graph
  - Shortest Path Problem
  - Dijkstra's Algorithm
    - Proof of Correctness
    - Complexity Analysis

Ch.4. Greedy Algorithms

(04mst.pdf) (04demo-mst)

- 4.5. Minimum Spanning Tree
  - Greedy Algorithms
    - Prim's algorithm
      - Implementation
      - Proof of Correctness
    - Kruskal's algorithm
      - Implementation
      - Proof of Correctness
    - Reverse-Delete algorithm
  - Cycles and Cuts
  - Cycle-Cut Intersection
  - Cut property
  - Cycle property
  - Prim's Algorithm: Proof of Correctness

Ch.4. Greedy Algorithms

(04mst.pdf, 04huffman.pdf)

- 4.7. Clustering
  - Clustering of Maximum Spacing
  - Greedy Clustering Algorithm
    - Single-link k-clustering algorithm.
      - Analysis
- 4.8. Huffman Codes
  - Data Compression
  - Prefix Codes
  - Optimal Prefix Codes
    - Average bits per letter (ABPL)
  - Representing Prefix Codes using Binary Trees
    - Full Tree
  - Huffman Encoding
    - Algorithm & Efficient implementation
    - Huffman Code Construction

Ch.5. Divide and Conquer

(05divide-and-conquer.pdf)

(05demo-merge.pdf, 05demo-merge-invert.pdf)

- 5.1. Mergesort
  - A Useful Recurrence Relation
  - Proof by Recursion Tree
  - Proof by Induction

- 5.3 Counting Inversions
  - Counting Inversions: Divide-and-Conquer
    - Implementation

- 5.4 Closest Pair of Points
  - 1D version: Divide and Conquer
  - 2D version: Divide and Conquer
    - Algorithm
    - Analysis

Ch.5. Divide and Conquer

(05multiply.pdf)

- 5.5. Integer Multiplication
  - Integer addition
  - Integer multiplication
  - Divide-and-Conquer Multiplication
    - Recursion Tree
  - Karatsuba Multiplication
    - Recursion Tree
  - Matrix Multiplication
    - Fast Matrix Multiplication: Warmup
      - Divide-and-Conquer
    - Fast Matrix Multiplication: Strassen
      - Key Idea