CSCE 311 - S24:
Final Study Guide

Prof. Edward Talmage

April 29, 2024

Learning Goals

This is an outline of the things you have learned so far. Problems on the exam will reference these, showing why these problems were selected to test your knowledge.

1. Math Background

    (a) Logarithms
    (b) Definition of Algorithm
    (c) Pseudocode
    (d) `InsertionSort` and Analysis

2. Proof Techniques

    (a) Direct
    (b) Contradiction (Indirect)
    (c) Contrapositive
    (d) Construction
    (e) Cases
    (f) Induction

3. Algorithmic Analysis

    (a) Asymptotics: $O, \Omega, \Theta, o, \omega$
        i. Usage for upper and lower bounds
        ii. Tight bound (Θ)
        iii. Strict bounds (little-o, little-ω)
        iv. Anonymous asymptotics
        v. Rules for combining functions, polynomials, hierarchy
    (b) Runtime Recurrences
        i. Recursion Trees
        ii. Substitution
        iii. Master Theorem

4. Divide & Conquer

    (a) Recursive and non-recursive parts of a D&C Algorithm
    (b) Optimal Substructure
    (c) Deriving a Runtime Recurrence
    (d) Using a recurrence to guide algorithmic improvements (Strassen's Algorithm)

5. Dynamic Programming

(a) Formal Outline

    (b) Bottom-Up vs. Top-Down/Memoization

    (c) Subproblem Graphs

    (d) Requirements: Optimal Substructure, Overlapping Subproblems

    (e) Problem Size and Pseudo-Polynomials

    (f) Examples: Fibonacci, Rod Cutting, Matrix Chain Multiplication, Longest Common Subsequences, etc.

6. Greedy Algorithms

    (a) Outline for Correctness Proofs

    (b) Requirements: Optimal Substructure, Greedy Choice Property

    (c) Examples: Activity Selection, Fractional Knapsack, Binary Character Encoding, Huffman Codes, Scheduling with Deadlines, etc.

7. Data Structures

    (a) Abstract Data Types

    (b) Data Structure Definition

    (c) Examples: Queue, Stack, Bounded Queue Implementation

    (d) Trees

        i. Tree Representations
        ii. Binary Search Trees
        iii. Red-Black Tree Overview
        iv. B-Trees

    (e) Disjoint Sets

8. Amortized Analysis

    (a) Usage

    (b) Methods of Amortized Analysis:

        i. Aggregate
        ii. Accounting
        iii. Potential

    (c) Examples: Queue from Two Stacks, Stack with Multipop, k-Bit Binary Counter, Dynamic Table, Disjoint Sets

9. Graph Algorithms

    (a) Representation: Dynamic, Adjacency List, Adjacency Matrix

    (b) BFS, DFS

    (c) Applications of DFS: Topological Sort, Strongly-Connected Components

    (d) Minimum Spanning Trees: Generic, Prim, Kruskal

    (e) Shortest Paths

        i. Problems (SSSP, APSP)
        ii. Bellman-Ford
        iii. Dijkstra
        iv. Floyd-Warshall (APSP algorithm from recitation)

    (f) Flow

        i. Residual Graphs, Augmenting Paths
        ii. Ford-Fulkerson Outline
        iii. Edmunds-Karp Implementation
        iv. Min-Cut, Max-Flow

10. Limits of Computation

    (a) P, NP, NP-Complete

    (b) Undecidability

        i. Combinatorial proof (infinities)

        ii. Halting Problem

    (c) Rice's Theorem