CPSC 335 (Wortman) - Summer 2018 - Final Study Guide

The exam will take place Wednesday June 27, in class. It will cover chapters 6-12 of the ADITA textbook, which corresponds to the material for weeks 3-5 in the syllabus.

The format of the final exam will be similar to that of the midterm exam, so will resemble exercises and quizzes. You may be asked to define terminology; write problem statements; prove efficiency classes; analyze pseudocode; execute the algorithms we have studied by hand, showing work; and design algorithms using exhaustive search, decrease-by-half, randomization, reduction, and dynamic programming. You may be asked to give lower bounds for problems, categorize problems into efficiency classes, and prove that problems fit into particular efficiency classes. You will not be asked to prove an NP-complete reduction.

You may bring a memory aid ("crib sheet") to the exam. Your sheet must be a single piece of paper, letter/A4 sized, optionally double-sided, with your name and CWID on it, and you must turn it in with your exam packet.

The following material is fair game:

- Chapter 6 Exhaustive Search and Optimization
    - 6.1: terminology of candidates, verification, and acceptable candidates
    - 6.2 Proper Exhaustive Search: the pattern
    - 6.3 Exhaustive Optimization: the pattern, and differences from proper search
    - 6.4 Designing and Analyzing Exhaustive Algorithms: the steps to design an algorithm and fill in the blanks, and the patterns for time complexity
    - 6.5 Generating Candidates: we skimmed this section; you should just know how to generate candidates using nested loops, a permutations function, and a subsets function
    - 6.6 Minimum Spanning Trees by Exhaustive Search: be able to follow how this algorithm works, how you could follow a similar process to design a different algorithm, and why its time complexity is exponential
- Chapter 7 Decrease-by-Half (Divide and Conquer)
    - 7.1 The Big Idea
    - 7.2 Example: Summation
    - 7.3 Analyzing Decrease-By-Fraction Algorithms: be prepared to prove the efficiency class of algorithms using the master method.
    - 7.4 Merge Sort: be able to run the algorithm be hand, and know its $O(n \log n)$ time bound which is important for the tight bound on sorting
    - 7.6 Indivisible Problems: notably circuit SAT seems to be impossible to solve with decrease-by-half
- Chapter 8 Randomization
    - 8.1 The Big Idea
    - 8.2 Generating Random Numbers
    - 8.3 The Monte Carlo Pattern

    - 8.4 The Las Vegas Pattern
    - 8.5 Quick Sort: we covered quick sort, but only skimmed the details of in-place partitioning
- Chapter 9 Reduction
    - 9.1 The Big Idea
    - 9.2 Reduction to Sorting: median finding, set intersection
    - 9.3 Reduction to Hash Table Operations: includes an overview of how hash tables work, and that their critical operations take $O(1)$ expected time
    - 9.4 Priority Search Queues; Revisiting Prim-Jarnik and Dijkstra: we skimmed this section; the takeaway is that these two algorithms can be sped up to $O(m + n \log n)$ time each using the Fibonacci heap data structure
- Chapter 10 Dynamic Programming
    - 10.1 The Big Idea
    - 10.2 1D Dynamic Programming and Fibonacci Numbers
    - 10.3 American Change-Making
    - 10.4 2D Dynamic Programming and Universal Change-Making
- Chapter 11 Lower Bounds
    - 11.1 The Big Idea
    - 11.2 Notation and Terminology: lower/upper/tight bounds, big-Omega, big-Theta
    - 11.3 Proving Negatives
    - 11.4 Trivial Lower Bounds
    - 11.5 The Tight Bound for Sorting
    - 11.6 Reduction Arguments: know how to prove a lower bound with a reduction argument
- Chapter 12 Intractable Problems
    - 12.1 The Big Idea
    - 12.2 Efficiently-Solvable Problems and P
    - 12.3 Unsolvable Problems and Decidability: undecidable problems exist; the halting problem
    - 12.4 Verifiable Problems and NP
    - 12.5 Hard Problems and NP-Hardness: we skimmed this section
    - 12.7 Proving NP-Completeness By Reduction: we skimmed this section
    - 12.8 P Versus NP Revisited: we skimmed this section