CS 362, Review

Jared Saia
University of New Mexico

Final

- 5 questions
- There will be some time pressure, so make sure you can solve problems both quickly and correctly.
- I expect a class mean between 50 :(and 70 :) points

How to Prepare

The best way to study is solving problems yourself. This is more effective than either watching me solve problems, or reading problem solutions in the slides and the textbook (although you should do these too). For sources of problems to solve, I'd focus on the following, in order of priority:

1. Problems from my past midterms and finals (CS362 and CS561)
2. Problems from our homeworks - make sure that you can re-solve the hw problems quickly and correctly. Also, try to resolve problems from lecture and in-class exercises.
3. Problems from the textbook, from other classes, online sources, etc., etc.

Topics Covered - Pre-Midterm

- Probability and Randomized Algorithms: Linearity of Expectation, Union Bounds, Markov's inequality. Randomized Quicksort, Skip Lists, MAX-3SAT Approximation.
- Recurrence Relations and Induction (Chapter 3 and 4 in text) : Big-O and friends, recursion trees, Master method, annihilators and change of variables; Proof by induction!
- Dynamic Programming: String Alignment, Matrix Multiplication, Longest Common Subsequence (Chapter 15)
- Greedy Algorithms: Activity selection, fractional knapsack, MST, proof via exchange property (Chapter 16)
- Amortized Analysis: Aggregate Method, Accounting Method, Potential Method, Dynamic Array (Chapter 17)

Topics Covered - Post-Midterm

- Disjoint-Sets: Union by Rank and Path Compression, Amortized Costs (Chapter 21)
- Minimum Spanning Trees: Kruskal's and Prim's Algorithm, Safe Edge Theorem and Corollary
- Shortest Paths: Dijkstra's, Bellman-Ford, Floyd-Warshall (Chapters 22 23,24,25)
- NP-Hard Problems: Definitions of P, NP, co-NP, NP-Hard, and NP-Complete; Reductions (i.e. how to show that a problem is NP-Hard); Classic NP-Hard problems: CIRCUIT-SAT, SAT, 3-SAT, COLORING, CLIQUE, VERTEX COVER, INDEPENDENT SET, HAMILTONIAN CYCLE, TSP. (Chapter 34)
- Approximation Algorithms: Vertex Cover, TSP, MAX-3SAT

Example Problem - Short Answer

Collection of true/false questions, matching and short answer questions. Some examples:

- T/F questions covering all topics
- Multiple Choice e.g. I give you some "real world" problems and ask you which algorithm we've studied in class that you would use to solve each of them; I give you some problems and ask you how fast they can be solved, etc.
- Know the resource bounds for all algorithms covered.

Example Problem - Induction/Recursion

- Proof by induction over a structure (e.g. tree, string) or algorithm
- Think "divide and conquer"
- In the IS: (1) make problem "smaller"; (2) Apply the IH; (3) "Paste up": use solution from IH to get solution to big problem

Example Problem - Dynamic Programming & Greedy Algorithms

- Key focus will be on getting the correct recurrence relation
- Probably related to some problem we did in class and/or homework
- Practice solving a big problem by using solutions to sub-problems (divide and conquer)
- Greedy algorithms: Exchange argument based proofs

Example Problem - Graph Theory

- Possibility 1: MST and Safe Edge theorem
- Possibility 2: Single Source Shortest Paths (Dijkstra's and Bellman-Ford)
- Possibility 3: All Pairs Shortest Paths

Example Problem - NP-Hardness

- Possibility 1: Something like Problem 1 from HW 7.
- Possibility 2: I give you a problem and ask you to prove it's NP-Hard by a reduction from another NP-Hard Problem.
- Possibility 3: I give you an NP-Hard Problem and ask you give an approximation algorithm for it (e.g. a variant of something already seen in class)