CS 561, Review

Jared Saia
University of New Mexico

Final

- About 5 main problems
- I recommend studying for this in advance, even though it is a take-home. The problems will require more than simple pattern-matching based on what we've done in class. If you go in cold, it will be challenging.
- As usual, the solutions will require lots of thinking, but not much writing.
- I expect a class mean of between 60 :(and 70 :) points

Topics Covered - Pre-Midterm

- Probability and Randomized Algorithms: Linearity of Expectation, Union Bounds, Markov's inequality. Randomized Quicksort, Count-min sketch, Bloom Filters.
- Recurrence Relations and Induction (Chapter 3 and 4 in text) : Defns of 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, SET-COVER, MAX-SAT.
- Linear Programming: Using it; Randomized Rounding.
- Gradient Descent: Basic, Online and Stochastic; Properties of convex/concave functions. Review the proofs!!!

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 - Review

Possibilities:

- Recurrence Relations
- Greedy Algorithms (Kruskal's and Prim's)
- Amortized Analysis
- Proof by Induction

Example Problem - Dynamic Programming

- 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

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)

Example Problem - Gradient Descent

- Regular
- Online
- Stochastic
- Problem Idea: Something like the Saia Trucking problem