COMPSCI 330 Design and Analysis of Algorithms

Final Review

Materials for the finals

- Final problems can be related to everything, except for
  - The bipartite matching algorithm (just the algorithm, not the problem).
  - Proofs for linear programs to have integer solutions.
  - Machine learning algorithms.

Once again, remember 3 techniques

- Divide and conquer
- Dynamic Programming
- Greedy Algorithms

- Everything is still related to these.

Divide and conquer

- Use when you can partition problems into unrelated subproblems.
- Midterm: Recursion trees, master theorem etc.
- Final: Remember analysis for randomized quicksort and quick selection?

Dynamic Programming

- Very general technique, can be combined with anything
- Dynamic programming can be combined with graphs/amortized analysis/probabilities
- Will see an example later.

Greedy

- Proof is crucial --- you will lose many points if you just give a greedy algorithm, even if it's correct.

- Greedy can also be combined with everything we talked about in the latter half.

Randomized algorithm

- Things to remember:

- Basic probabilities
- Conditioning and independence
- Law of total expectation/Analyzing expected running time.

Amortized analysis

- Definition

- Techniques
  - Aggregate
  - Accounting
  - Potential

- Amortized cost analysis for union find will not appear.

Linear Program

- Definition
- Primal and Dual
- How to use a LP to solve a problem?

Reductions

- Direction of reduction

- Basic problems: 3-SAT, INDEPENDENT SET, HAMILTONIAN PATH, TRIPARTITE MATCHING.

Sample Problems

Election prediction

- You are trying to predict the result of election based on available data. There are n states, state i has v_i votes, and there are m votes in total. Based on the current data, state i has probability p_i to vote for democratic party, and probability $1-p_i$ to vote for republican party. Compute the exact probability of democratic party to win the election.

Making a Heap

- Given a complete binary tree with n vertices. Design an algorithm that makes this tree a heap.
- Your algorithm should be as fast as possible.

Traffic Routing

- Suppose there is a road network represented as a graph. Each edge has a capacity which shows the number of cars it can transport at a unit time (in both directions). There is now a basketball game, and there are a_i people from neighborhood i that wants to go to some parking lot. Each parking lot has capacity b_i. Assume driving is actually very fast as long as it's not congested, compute the amount of time it takes to transport all these people.
- Write a linear program to solve this problem.

Quadratic Equations

- We can solve linear equations and linear inequalities (LP). However, suppose we are given m quadratic equations over n variables, and we would like to decide whether they have a solution.

- Based on the NP-completeness of 3-SAT, prove QUADRATIC EQUATION is also NP-complete.

Final Format

- 5 problems + 1 extra. 20 points each. 3 hours
- Difficulty is similar to midterm.
- Extra problem is very hard. Do not attempt until you are confident with your other answers. (and no partial credit is given for the extra problem.)