CS 314: Algorithms, Spring 2009

Midterm Exam — April 24, 2009

Name:	Email Address:

1. This is a closed-book and closed-notes exam. You are allowed one "cheat sheet" on standard 8.5 by 11 inch paper, handwritten on the front and back.
2. Print your full name and your email address in the boxes above.
3. Print your name at the top of every page.
4. Please write clearly and legibly. If I can't read your answer, I can't give you credit.
5. When asked to design an algorithm, you must also provide a runtime analysis for that algorithm. However, proofs of correctness are NOT required for algorithms on this exam. However, problems that explicitly ask you to prove something still require you to do proofs!
6. There are 4 problems on the exam. Your grade will be calculated based only on 3 out of the 4 problems. Please feel free to try all of them, though; I'll take the maximum of the 3 for your final grade.
7. As with any problem, you are welcome to write "I don't know" **and nothing else** in order to receive 25% of the total points. This only applies to entire problems, however; for example, you may not give an algorithm but not compute the runtime and still say "I don't know" for just the runtime portion of the problem.
8. Remember, these are NOT necessarily in order of difficulty. Please read all the problems first, and don't allow yourself to get stuck on a single problem.

#	1	2	3	4	Total
Max	20	20	20	20	60
Score					

1. Suppose you are given an undirected graph $G = (V, E)$ with nonnegative weights w_e on each edge $e \in E$; you are also given a tree T which is the minimum spanning tree of G.

   Now assume that an edge e is added to the graph G. Give an algorithm to test if T is still the minimum spanning tree of G plus e; if it is no longer the minimum spanning tree, you must compute the new spanning tree of G plus e.

   For full credit, your algorithm must run in $O(n + m)$ time, where $n = |V|$ and $m = |E|$.

2. Suppose you rent a room in a co-op house with $n-1$ other people. Each night, one of you is supposed to cook dinner for the others. However, each of you has scheduling constraints - nights when you can't cook because you have to work or have an exam the next day, for example. Your goal is to design a schedule so that each of you cooks the same number of nights and no cook is scheduled to cook on a night when they are busy.

   More formally, we have a list of people $p_1, \ldots, p_n$, a list of days $D = \{d_1, \ldots, d_k\}$, and each person has a conflict set $S_i \subseteq D$ of days when they can NOT cook dinner. Design an algorithm to find a schedule (if one exists) so that everyone cooks the same number of days, and no one cooks on a day they are busy - in other words, if person p_i cooks on day d_j, then $d_j \notin Si$.

   (You may assume that $n > k$ and n is divisible by k, so that everyone will cook exactly the same number of nights.)

3. Given two graphs G and H, the subgraph isomorphism problems asks if G contains an exact copy of H.

   Prove that subgraph isomorphism is NP-Complete.

4. Suppose you want to take a road trip from St. Louis to New York via a network of roads. You don't care how long it takes, how many stops there are, or how much gas you use - you just want to minimize the amount of money you spend on food. You pick a single restaurant to stop at in each city ahead of time, so you know exactly the dollar amount you will spend on food if you stop at a certain city.

   More formally, you are given a directed graph $G = (V, E)$ with nonnegative *vertex* weights $w : V \to \mathbb{R}^+$, a source vertex s, and a target vertex v. Describe and analyze an efficient algorithm to find a minimum weight path from s to t.

   (Hint: Modify the graph somehow and run an algorithm we already know!)

(scratch paper)

(scratch paper)