CSE 830: Design and Theory of Algorithms

Homework #5: Due Tuesday, December 6th 2011, 10:20am

1. The **Graph-isomorphism problem** is an open problem; no one has yet been able to prove it to be a hard problem, yet no polynomial time algorithm has been constructed either. Show that Graph Isomorphism is in NP by describing what certificate should be returned with a "yes" answer, and how that certificate can be verified in polynomial time. Analyze the time complexity for this verification.

2. The **Subgraph-isomorphism problem** is much easier to prove hard than its full-graph counterpart. This problem takes two graphs G_1 and G_2 and asks whether G_1 can be found as a subgraph in G_2. Show that the subgraph-isomorphism problem is NP-complete. (*Hint*: Consider a reduction from Maximum Clique or Hamiltonian Cycle.)

3. ***Implement*** an algorithm to solve the *vertex cover* problem. To do this, use the Minimum Dominating Set program implemented by one of your group members for the class project. Write only two additional sections in your program: one to pre-process the input graph to run under the Minimum Dominating Set, and one to post-process the solution and return the minimum vertex cover. Both pre- and post-processing should be polynomial time algorithms. Hand in only a printout of the new sections of your code, and let me know whose program you used as a base. I may ask you for an electronic copy of the program if I want to test it.

4. 2-SAT: In class, I claimed that the 2-satisfiability problem could be solved in polynomial time. Create an algorithm that will do so! (*Hint*: Observe that $x \vee y$ is equivalent to $\neg x \rightarrow y$ and $\neg y \rightarrow x$. You can reduce 2-SAT to a problem on a directed graph that is efficiently solvable.)

5. Book problem: 34-3 (Graph Coloring)

6. *(extra credit)* **Downsizing:** MegaCorp achieved a remarkable growth rate for a number of years, and spawned new departments too rapidly. The board of directors decided to shrink the company, and want it to match the structure of their leaner competitor EarthTech Inc. You are given a list of all departments (vertices) and which pairs of departments can be merged (edges). When departments merge, the new department must perform all functions of the original pair (and thus maintain all connections). You must find a collection of department mergers that will make the new layout of MegaCorp be *isomorphic* to EarthTech. Give a polynomial time algorithm to solve this *or* prove the problem NP-Complete.