Homework 3, CS520, Spring '09, McConnell

Due Thursday, 3/12

1. We studied a problem where you are a trucker who shows up at a loading dock with a set of items to choose from, and you want to maximize your load subject to the constraint that you can't go over a certain limit. We showed that there is no polynomial-time algorithm for this problem unless P=NP.

   Our strategy was to come up with a corresponding decision problem. We showed that a polynomial algorithm for the optimization problem could be used to solve this decision problem in polynomial time. We then showed that the decision problem in NP, and finally, that it was NP-complete, by reducing a known NP-complete problem to it.

   Consider a variant of the problem where each item carries a *profit* that is not necessarily proportional to the weight. The goal is to maximize the total profit of your load, subject to the given weight limit. Show that this new problem has no polynomial-time solution unless P = NP.

2. A variation on the Hamiltonian Cycle problem is that of determining whether there is a path that visits all of the vertices without backtracking through any vertex. Unlike the original problem, the path is not required to be a cycle, that is, it can start at one vertex and end at another.

   Prove that the new problem is NP-complete.

3. You are given network of computers, represented by a graph, where the computers are vertices and two vertices are adjacent if the two computers have a direct link between them.

   Your boss wants you to set up a way to route messages between computers of the network. Your predecessor thought it would be a good idea to use a minimum spanning tree. The owner of one of the computers complained that this was an incompetent solution, because his computer had much higher degree than any of the others, and it was spending all of its time passing along other people's messages.

   Your boss wants you to find a spanning tree that minimizes the maximum degree, making the solution as fair as possible. Show that this problem is NP-hard.

   You may make use of any NP-completeness results from the book or from the above problems, even if you didn't solve them.