SOLUTIONS, Homework 3, CS520, Spring '09, McConnell

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.

   Solution: *A corresponding decision problem is the question of whether you can achieve a given profit p. A polynomial-time algorithm for finding the highest-achievable profit could be used to solve this decision problem in polynomial time.*

   This decision problem is in NP, since it takes polynomial time to check whether a proposed solution achieves profit p without exceeding the weight constraint. Such a solution always exists if the answer to the decision problem is yes.

   It's NP-complete because Subset-Sum, which is NP-complete, is just the special case where the profits are equal to the weights. A polynomial-time solution to the optimization would give a polynomial-time solution to this decision problem, which would imply that $P = NP$.

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.

   Solution: *The problem is in NP, since it takes polynomial time to check whether a proposed Hamiltonian path is legitimate.*

   There is a Hamiltonian cycle in a graph G if and only if, for each vertex v, there is a Hamiltonian path whose endpoints are at v and a neighbor w of v.

   Select v and add a new vertex v' adjacent only to v. Add a new vertex w' adjacent to all neighbors of v and a new vertex w'' adjacent only to w. Call this new graph G'. Since v' and w'' have degree 1, any Hamiltonian path in G' begins and ends at v' and w''. There is such a path in G' if in G there is a Hamiltonian path whose endpoints are at v and a neighbor of v, hence, if and only if G has a Hamiltonian cycle.

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 find one of the results from the previous problem to be helpful.

   Solution: *A corresponding decision problem is whether there exists a spanning tree with maximum degree k. This problem is in NP, since it takes polynomial time to evaluate whether a proposed spanning tree is, in fact, a spanning tree has has maximum degree at most k.*

   Hamiltonian Path, which is NP-complete, is just the special case where $k = 2$. Since Hamiltonian Path is NP-complete (by the result of the above problem), so is this one.

   A polynomial-time solution to the optimization problem would give a polynomial-time solution to the decision problem, which would imply that P = NP.