Algorithms
Computer Science 140 & Mathematics 168
Spring 2007
Homework 10b
Due Tuesday, April 10

- *Reminder:* The second exam will be given out on Thursday, April 12 in class. You may choose any 24 hour block between Thursday, April 12 and Tuesday, April 17 to take the exam. The exam will be due back no later than Tuesday, April 17 in class. As with the last exam, you may use your lecture notes, homework solutions, and solution sets that I have provided. No other resources (e.g. books, electronic resources, oracles) are permitted.

- *This homework set is challenging. Please start early!*

- It is recommended that you work on these problems in the order that they are given. The solution to a given problem may rely on results from previous problems. If you skip a problem, at least be aware of the result that you would have shown in that problem so that you can use that result in a subsequent problem.

- Try the bonus problems if you have time. They will significantly strengthen your NP-completeness muscles and they are fun and interesting!

- Please be sure to read the document that I've written on writing NP-completeness proofs. This will remind you of what's necessary in such a proof and serves as a good example of the level of precision required.

1. **[10 Points] Professor Lai and NP!**

   Professor I. Lai from P.I.T. claims that if P_1 and P_2 are any two problems in NP such that $P_1 \leq_p P_2$, then $P_2 \leq_p P_1$. Prove that this would imply that P = NP.

2. **[15 Points] ILP Revisited!**

   In class we proved that Integer Linear Programming (ILP) is NP-complete. Our reduction was from 3SAT. In this problem you will prove that ILP is NP-complete using a reduction from Vertex

COVER. You do not need to show that ILP is in NP (we already argued that in class).

3. **[15 Points] Hitting Set!** First, the gratuitous story: The Computer Science Department at The Western Institute of Technology (T.W.I.T.) has 5 faculty members, Dr. Theo Rize, Dr. I. Dutu, Dr. Sam Heer, Dr. Mia Thue, and Dr. Juan More. Each professor serves on a number of committees. The Polynomial Time Algorithms Committee consists of Professors Rize, Dutu, and More. The NP-Completeness Committee consists of Professors Dutu, Heer, and Thue. Finally, the Other Stuff Committee consists of Professors Thue and More.

   The Dean would like to convene a meeting in which the number of computer scientists invited is minimized (she can't stand their bad CS jokes!) but each committee is represented by at least one of its members. For example, inviting Professors Thue and More would be a solution in this case.

   The Dean has formulated this problem as the following general problem (known to computer scientists as the HITTING SET Problem): We are given a set S (the set of professors) and a collection, C, of subsets of S (the set of committees).

   A Hitting Set for C is a subset $S' \subseteq S$ such that S' contains at least one element from each subset in C. The HITTING SET optimization problem is to find the smallest Hitting Set.

   (a) Formulate the HITTING SET decision problem corresponding to the optimization problem described above.

   (b) Prove that the HITTING SET decision problem that you formulated is NP-complete via a reduction from a problem that we've shown in class to be NP-complete.

4. **[20 Points] Dominating Set.** The DOMINATING SET problem is stated as follows: Given a graph $G = (V, E)$ and a positive integer ℓ, does there exist a subset S of ℓ vertices in V such that every vertex in V is either in S or is connected by an edge to some vertex in S.

   Take a close look at this problem to make sure that you understand how this problem differs from the seemingly similar VERTEX COVER problem. Prove that DOMINATING SET is NP-complete using a reduction from 3SAT.

5. [**15 Points**] **The Rural Fire Station Problem!** Let $G = (V, E)$ be an undirected graph in which the vertices represent small towns and the edges represent roads between those towns. Each edge e has a positive integer weight $d(e)$ associated with it, indicating the length of that road. The *distance* between two vertices (towns) in a graph is defined to be the length of the shortest weighted path between those two vertices.

   Each vertex v also has a positive integer $c(v)$ associated with it, indicating the cost to build a fire station in that town.

   In addition, we are given two positive integer parameters D and C. Our objective is to determine whether or not there is a way to build fire stations such that the total cost of building the fire stations does not exceed C and the distance from any town to a fire station does not exceed D. This problem is known as the RURAL FIRE STATION (RFS) Problem.

   Your company has been hired by the American League of Rural Fire Departments to study this problem. After spending months trying unsuccessfully to find an efficient algorithm for the problem, your boss has a hunch that the problem is NP-complete. Prove that RFS is NP-complete using a reduction from a problem that we have already shown to be NP-complete.

6. [**15 Point OPTIONAL Bonus Problem!**] **Clique Revisited!**

   In class we proved that CLIQUE is NP-complete via a reduction from INDEPENDENT SET or VERTEX COVER. Your task here is to prove that CLIQUE is NP-complete by a simple reduction directly from 3SAT. (I say "simple" just to avoid using a reduction that is the composition of reductions that we have already seen.)

   If you prefer, you may come and talk to me in person to show me your proof on my whiteboard. This will save you time in writing up your solution and will still allow you to claim your bonus points!

7. [**20 Point OPTIONAL Bonus Problem!**] **Multicommodity Network Flow!**

We know how to solve the network flow problem in polynomial time. A generalization of network flow is called *multicommodity network flow* and it goes like this: We are given a flow network except now there are multiple source/sink pairs in the network. That is, there are some number k of sources $s_1, \ldots, s_k$ and the same number of sinks $t_1, \ldots, t_k$. Each source s_i pushes a different kind of "fluid" or "commodity" to its corresponding sink t_i. However, the edges of the network are shared by these flows and each edge still has a single integer capacity. For example, an edge with capacity 5 might accommodate 2 units of flow from s_1 to t_1 and 3 units of flow from s_2 to t_2. Now we are interested in finding a set of flows, one for each source/destination pair. Of course, we are still subject to capacity constraints and conservation of flow (i.e. for any given commodity, the amount of flow for that commodity entering a given vertex is equal to the amount of flow for that commodity leaving that vertex).

It turns out that maximizing the total amount of flow in a multicommodity network can still be solved in polynomial time by using polynomial time linear programming algorithms! However, the optimal solution found by the linear program will, in general, not have integer flows.

If we now simply restrict the flows for each commodity to have integer values, the problem becomes NP-complete! Amazing, but true! In particular, the decision problem that we will consider here is this: Given is a flow network with k commodities and k source/sink pairs, (s_i, t_i) (one for each commodity), and a demand d_i for each sink t_i. The question is: "Does there exist a valid set of flows, one for each commodity, such that each sink receives d_i of commodity i?"

Prove that this problem is NP-complete. A reduction from 3SAT is particularly nice! (It turns out that this problem remains NP-complete even if there are just two source/sink pairs and thus two commodities, and each edge has capacity 1. You don't need to prove this stricter result, but it's kind of surprising that NP-completeness is maintained even under these very tight constraints!)

If you prefer, you may come and talk to me in person to show me your proof on my whiteboard. This will save you time in writing up your solution and will still allow you to claim your bonus points!