Due at the start of class Tuesday, December 11, 2007.

We know a number of problems are NP-complete including: Circuit SAT, SAT, 3-SAT, Independent Set, Vertex Cover, Set Cover, Hamiltonian Cycle, Hamiltonian Path,, Traveling Salesman, 3-Dimensional Matching, Graph Coloring, Subset Sum, Clique, and Subgraph Isomorphism.

Problem 1. HAMILTONIAN PATH PROBLEM: given a directed simple graph, does it contain a path that starts at some vertex and goes to some other vertex, going through each remaining vertex exactly once.

HAMILTONIAN CYCLE PROBLEM: given a directed simple graph, does it contain a directed simple cycle that goes through each vertex exactly once.

The book uses the HAMILTONIAN CYCLE PROBLEM to prove that the HAMILTONIAN PATH PROBLEM is NP-complete.

Assume that the HAMILTONIAN PATH PROBLEM is known to be NP-complete. Given this assumption, prove that the HAMILTONIAN CYCLE PROBLEM is NP-complete. (Make sure to show that the HAMILTONIAN CYCLE PROBLEM is in NP.)

Problem 2. Consider the problem DENSE SUBGRAPH: Given G, does it contain a subgraph H that has exactly K vertices and at least Y edges? Prove that this problem is NP-complete.

Problem 3. Assume that the following problem is NP-complete.
PARTITION: Given a finite set A and a "size" $s(a)$ (a positive integer) for each $a \in A$. Is there a subset $A' \subseteq A$ such that $\sum_{a \in A'} s(a) = \sum_{a \in A-A'} s(a)$?

Now prove that the following SCHEDULING problem is NP-complete:
Given a set X of "tasks", and a "length" $\ell(x)$ for each task, three processors, and a "deadline" D. Is there a way of assigning the tasks to the three processors such that all the tasks are completed within the deadline D? A task can be scheduled on *any* processor, and there are no precedence constraints. (Hint: first prove the NP-completeness of SCHEDULING with two processors.)

Problem 4. Prove that the following ZERO CYCLE problem is NP-complete:
Given a simple directed graph $G = (V, E)$, with positive and negative weights $w(e)$ on the edges $e \in E$. Is there a simple cycle of zero weight in G ? (Hint: Reduce PARTITION to ZERO CYCLE.)

Problem 5. The WEIGHTED 3-DIMENSIONAL MATCHING PROBLEM is the same as 3-DIMENSIONAL MATCHING except each triple has a weight. In the optimization version the goal is to find a 3-dimensional matching with the maximum possible weight.

(a) Define a decision version of the WEIGHTED 3-DIMENSIONAL MATCHING PROBLEM.

(b) Show that the decision version is in NP.

(c) Show that the decision version is complete for NP (that is, it is NP-hard).

(d) Show that if you could solve the optimization version in polynomial time that you could also solve the decision version in polynomial time.

(e) Show that if you could solve the decision version in polynomial time that you could also solve the optimization version in polynomial time. HINT: First find the weight of an optimal matching.

Problem 6. Do Exercise 7 on page 507 of Kleinberg and Tardos.