For all reductions, be sure to explain why it is polynomial-time.

1. Fill out the course evaluation and then copy and sign the following statement in your homework: “I have filled out the 451/Spring 2015 course evaluation.”

2. (20 pts)
   
   (a) Suppose we know that problem \( X \) is NP-complete. Suppose we discover a polynomial time algorithm for \( X \). Would that imply that the SATISFIABILITY problem can be solved in polynomial time? Explain your answer.

   (b) Suppose we know that problem \( X \) belongs to NP. Suppose we discover a polynomial time algorithm for \( X \). Would that imply that the SATISFIABILITY problem can be solved in polynomial time? Explain your answer.

   (c) Suppose we discover an \( O(n^3) \) algorithm for SATISFIABILITY. Would that imply that every problem in NP can be solved in \( O(n^3) \) time? Why or why not?

3. (20 pts) The HALF-CLIQUE problem is: given an undirected graph \( G = (V, E) \), does there exist a clique of size exactly \( \frac{n}{2} \)? Prove that HALF-CLIQUE is NP-complete via the following steps:

   (a) Show that HALF-CLIQUE is in NP.

   (b) Give a polynomial-time reduction from CLIQUE of HALF-CLIQUE. Prove that your reduction is correct and takes polynomial time to construct.

4. (20 pts) Suppose you have access to an oracle algorithm \( ORACLE_{HC} \) that, given any directed graph \( G \), tells you whether \( G \) has a Hamiltonian cycle. (A Hamiltonian cycle is one that visits each vertex exactly once.) Assume that a single call to \( ORACLE_{HC} \) takes \( O(f(n, m)) \) time, where \( f(n, m) \) is a polynomial function in \( n \) and \( m \). Design an algorithm that, given any directed graph \( G \), determines whether \( G \) has a Hamiltonian cycle, and if it does, outputs it. Prove your algorithm is correct and give time complexity analysis.

5. (20 pts) Suppose you are given an undirected graph \( G = (V, E) \) and an integer \( k \), and you want to determine whether there are at most \( k \) vertices whose removal from \( G \) will destroy all the cycles in \( G \). Prove that this problem is NP-complete.

6. (Ungraded.) The 3-DIMENSIONAL MATCHING problem is a generalization of the bipartite matching problem: given disjoint sets \( X, Y, \) and \( Z \), each of size \( n \), and given a set \( T \subseteq X \times Y \times Z \) of triples, does there exist a set of \( n \) triples in \( T \) so that each element of \( X \cup Y \cup Z \) is contained in exactly one of these triples? The 3-DIMENSIONAL MATCHING problem is NP-complete. (See KT 8.6 for details.)

Consider the problem of assigning men, women, dogs and cars to households. There are \( n \) men, \( n \) women, \( n \) dogs and \( n \) cars. You are given as input a set \( A \) of 4-tuples, each
4-tuple consisting of a man, woman, dog and car that can form a single household. Is there a way to pick $n$ tuples from $A$ so that every man, woman, dog and car is contained in exactly one of them? (I.e., everyone and everything is assigned to a household, and no one and nothing is assigned to more than one household.) Prove that this problem is NP-complete.