**CPSC-320 Intermediate Algorithm Design and Analysis Winter 2000**

**Homework # 5**

Due in class on Friday, December 1.

Reminder: put, at the top left corner of your homework your name (last name first), your student ID number, and finally your tutorial section. Put each of these on a separate line. Below this information, include information on your collaborations, if any.

1. Problem 36.1, page 961, part (a) of Cormen, Leiserson, and Rivest:

   An *independent set* of a graph $G = (V, E)$ is a subset $V' \subset V$ of vertices such that each edge in $E$ is incident on at most one vertex in $V'$. The *independent set problem* is to find a maximum-size independent set in $G$.

   (a) Formulate a related decision problem for the independent set problem and show that your decision problem is NP-complete. Here, you need to show that your decision problem is in NP, and find a polynomial-time reduction from a problem that you already know to be NP-complete to your decision problem.

2. Problem 36.1, page 961, parts (b) and (c) of Cormen, Leiserson, and Rivest:

   (b) Suppose that you are given a subroutine to solve the decision problem you define in part (a). Give an algorithm that, given as input an undirected graph, uses this subroutine to find a maximum independent set for that graph. The running time of your algorithm should be polynomial in $|V|$ and $|E|$, where queries to the subroutine are counted as a single step.

   (c) Although the independent set problem is NP-complete, certain special cases are polynomial-time solvable. Give an efficient algorithm to solve the independent set problem when each vertex in $G$ has degree 2.

3. Suppose that $\Pi_1$ and $\Pi_2$ are decision problems and that there is a polynomial time reduction from $\Pi_1$ to $\Pi_2$. For each of the following questions, say whether the answer is true or false, and give a brief explanation of your answer.

   (a) If $\Pi_1$ is in P, must $\Pi_2$ be in P?

   (b) If $\Pi_2$ is in P, must $\Pi_1$ be in P?

   (c) If there is also a polynomial time reduction from $\Pi_2$ to $\Pi_1$, must $\Pi_1$ and $\Pi_2$ both be NP-complete?

   (d) If $\Pi_1$ and $\Pi_2$ are NP-complete, must there be a poly-time reduction from $\Pi_2$ to $\Pi_1$?

4. Exercise 37.2-1, Page 973, Cormen, Leiserson, and Rivest:

   (a) Show how in polynomial time we can transform one instance of the traveling salesman problem into another instance whose cost function satisfies the triangle inequality. The two instances must have the same set of optimal tours.

   (b) Explain why the polynomial-time transformation does not contradict Theorem 37.3 of the text, assuming that P is not equal to NP.