CS 373: Combinatorial Algorithms, Fall 2000
Homework 6 (due December 7, 2000 at midnight)

Name:		
Net ID:	Alias:	U $^3\!/\!_4$ 1

Name:		
Net ID:	Alias:	U $^3\!/\!_4$ 1

Name:		
Net ID:	Alias:	U $^3\!/\!_4$ 1

Starting with Homework 1, homeworks may be done in teams of up to three people. Each team turns in just one solution, and every member of a team gets the same grade. Since 1-unit graduate students are required to solve problems that are worth extra credit for other students, **1-unit grad students may not be on the same team as 3/4-unit grad students or undergraduates.**

Neatly print your name(s), NetID(s), and the alias(es) you used for Homework 0 in the boxes above. Please also tell us whether you are an undergraduate, 3/4-unit grad student, or 1-unit grad student by circling U, $^3\!/\!_4$, or 1, respectively. Staple this sheet to the top of your homework.

Required Problems

1. (a) Prove that P $\subseteq$ co-NP.
   (b) Show that if NP $\neq$ co-NP, then *no* NP-complete problem is a member of co-NP.

2. 2SAT is a special case of the formula satisfiability problem, where the input formula is in conjunctive normal form and every clause has at most *two* literals. Prove that 2SAT is in P.

3. Describe an algorithm that solves the following problem, called 3SUM, as quickly as possible: Given a set of n numbers, does it contain three elements whose sum is zero? For example, your algorithm should answer TRUE for the set $\{-5, -17, 7, -4, 3, -2, 4\}$, since $-5+7+(-2) = 0$, and FALSE for the set $\{-6, 7, -4, -13, -2, 5, 13\}$.

4. (a) Show that the problem of deciding whether one undirected graph is a subgraph of another is NP-complete.

   (b) Show that the problem of deciding whether an unweighted undirected graph has a path of length greater than k is NP-complete.

5. (a) Consider the following problem: Given a set of axis-aligned rectangles in the plane, decide whether any point in the plane is covered by k or more rectangles. Now also consider the CLIQUE problem. Describe and analyze a reduction of one problem to the other.

   (b) Finding the largest clique in an arbitrary graph is NP-hard. What does this fact imply about the complexity of finding a point that lies inside the largest number of rectangles?

6. *[This problem is required only for graduate students taking CS 373 for a full unit; anyone else can submit a solution for extra credit.]*

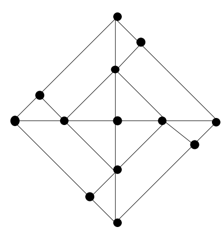
   PARTITION is the problem of deciding, given a set $S = \{s_1, s_2, \ldots, s_n\}$ of numbers, whether there is a subset T containing half the 'weight' of S, *i.e.*, such that $\sum T = \frac{1}{2} \sum S$. SUBSETSUM is the problem of deciding, given a set $S = \{s_1, s_2, \ldots, s_n\}$ of numbers and a target sum t, whether there is a subset $T \subseteq S$ such that $\sum T = t$. Give two reductions between these two problems, one in each direction.

Practice Problems

1. What is the *exact* worst case number of comparisons needed to find the median of 5 numbers? For 6 numbers?

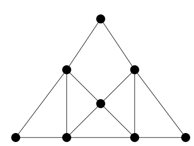
2. The EXACTCOVERBYTHREES problem is defined as follows: given a finite set X and a collection C of 3-element subsets of X, does C contain an *exact cover* for X, that is, a subcollection $C' \subseteq C$ where every element of X occurs in exactly one member of C'? Given that EXACTCOVERBYTHREES is NP-complete, show that the similar problem EXACTCOVERBYFOURS is also NP-complete.

3. Using 3COLOR and the 'gadget' below, prove that the problem of deciding whether a planar graph can be 3-colored is NP-complete. *[Hint: Show that the gadget can be 3-colored, and then replace any crossings in a planar embedding with the gadget appropriately.]*



Crossing gadget for PLANAR3COLOR.

4. Using the previous result, and the 'gadget' below, prove that the problem of deciding whether a planar graph with no vertex of degree greater than four can be 3-colored is NP-complete. *[Hint: Show that you can replace any vertex with degree greater than 4 with a collection of gadgets connected in such a way that no degree is greater than four.]*



Degree gadget for DEGREE4PLANAR3COLOR

5. Show that an algorithm that makes at most a constant number of calls to polynomial-time subroutines runs in polynomial time, but that a polynomial number of calls to polynomial-time subroutines may result in an exponential-time algorithm.

6. (a) Prove that if G is an undirected bipartite graph with an odd number of vertices, then G is nonhamiltonian. Give a polynomial time algorithm algorithm for finding a hamiltonian cycle in an undirected bipartite graph or establishing that it does not exist.
   (b) Show that the hamiltonian-path problem can be solved in polynomial time on directed acyclic graphs.
   (c) Explain why the results in previous questions do not contradict the fact that both HAMILTONIANCYCLE and HAMILTONIANPATH are NP-complete problems.

7. Consider the following pairs of problems:

(a) MIN SPANNING TREE and MAX SPANNING TREE
(b) SHORTEST PATH and LONGEST PATH
(c) TRAVELING SALESMAN and VACATION TOUR (the longest tour is sought).
(d) MIN CUT and MAX CUT (between s and t)
(e) EDGE COVER and VERTEX COVER
(f) TRANSITIVE REDUCTION and MIN EQUIVALENT DIGRAPH

(All of these seem dual or opposites, except the last, which are just two versions of minimal representation of a graph.) Which of these pairs are polytime equivalent and which are not?

★8. Consider the problem of deciding whether one graph is isomorphic to another.

(a) Give a brute force algorithm to decide this.
(b) Give a dynamic programming algorithm to decide this.
(c) Give an efficient probabilistic algorithm to decide this.
★(d) Either prove that this problem is NP-complete, give a poly time algorithm for it, or prove that neither case occurs.

*9. Prove that PRIMALITY (Given n, is n prime?) is in NP $\cap$ co-NP. Showing that PRIMALITY is in co-NP is easy. (What's a certificate for showing that a number is composite?) For NP, consider a certificate involving primitive roots and recursively their primitive roots. Show that this tree of primitive roots can be checked to be correct and used to show that n is prime, and that this check takes polynomial time.

10. How much wood would a woodchuck chuck if a woodchuck could chuck wood?