CS 473G: Combinatorial Algorithms, Fall 2005
Homework 1

Due Tuesday, September 13, 2005, by midnight (11:59:59pm CDT)

Name:	
Net ID:	Alias:

Name:	
Net ID:	Alias:

Name:	
Net ID:	Alias:

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.

Neatly print your name(s), NetID(s), and the alias(es) you used for Homework 0 in the boxes above. Staple this sheet to the top of your answer to problem 1.

There are two steps required to prove NP-completeness: (1) Prove that the problem is in NP, by describing a polynomial-time verification algorithm. (2) Prove that the problem is NP-hard, by describing a polynomial-time reduction from some other NP-hard problem. Showing that the reduction is correct requires proving an if-and-only-if statement; don't forget to prove both the "if" part and the "only if" part.

Required Problems

1. Some NP-Complete problems

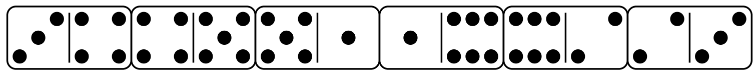
   (a) Show that the problem of deciding whether one graph is a subgraph of another is NP-complete.

   (b) Given a boolean circuit that embeds in the plane so that no 2 wires cross, PlanarCircuitSat is the problem of determining if there is a boolean assignment to the inputs that makes the circuit output true. Prove that PlanarCircuitSat is NP-Complete.

   (c) Given a set S with $3n$ numbers, 3partition is the problem of determining if S can be partitioned into n disjoint subsets, each with 3 elements, so that every subset sums to the same value. Given a set S and a collection of three element subsets of S, X3M (or *exact 3-dimensional matching*) is the problem of determining whether there is a subcollection of n disjoint triples that exactly cover S.

   Describe a polynomial-time reduction from 3partition to X3M.

(d) A *domino* is a 1×2 rectangle divided into two squares, each of which is labeled with an integer.[1] In a *legal arrangement* of dominoes, the dominoes are lined up end-to-end so that the numbers on adjacent ends match.



A legal arrangement of dominos, where every integer between 1 and 6 appears twice

Prove that the following problem is NP-complete: Given an arbitrary collection D of dominoes, is there a legal arrangement of a subset of D in which every integer between 1 and n appears exactly twice?

2. Prove that the following problems are all polynomial-time equivalent, that is, if *any* of these problems can be solved in polynomial time, then *all* of them can.

   - Clique: Given a graph G and an integer k, does there exist a clique of size k in G?
   - FindClique: Given a graph G and an integer k, find a clique of size k in G if one exists.
   - MaxClique: Given a graph G, find the size of the largest clique in the graph.
   - FindMaxClique: Given a graph G, find a clique of maximum size in G.

3. Consider the following problem: Given a set of n points in the plane, find a set of line segments connecting the points which form a closed loop and do not intersect each other.

   Describe a linear time reduction from the problem of sorting n numbers to the problem described above.

4. In graph coloring, the vertices of a graph are assigned colors so that no adjacent vertices recieve the same color. We saw in class that determining if a graph is 3-colorable is NP-Complete.

   Suppose you are handed a magic black box that, given a graph as input, tells you *in constant time* whether or not the graph is 3-colorable. Using this black box, give a *polynomial-time* algorithm to 3-color a graph.

5. Suppose that Cook had proved that graph coloring was NP-complete first, instead of CircuitSAT. Using only the fact that graph coloring is NP-complete, show that CircuitSAT is NP-complete.

[1] These integers are usually represented by pips, exactly like dice. On a standard domino, the number of pips on each side is between 0 and 6; we will allow arbitrary integer labels. A standard set of dominoes has one domino for each possible unordered pair of labels; we do not require that every possible label pair is in our set.

Practice Problems

1. Given an initial configuration consisting of an undirected graph $G = (V, E)$ and a function $p : V \to \mathbb{N}$ indicating an initial number of pebbles on each vertex, PEBBLE-DESTRUCTION asks if there is a sequence of pebbling moves starting with the initial configuration and ending with a single pebble on only one vertex of V. Here, a pebbling move consists of removing two pebbles from a vertex v and adding one pebble to a neighbor of v. Prove that PEBBLE-DESTRUCTION is NP-complete.

2. Consider finding the median of 5 numbers by using only comparisons. What is the *exact* worst case number of comparisons needed to find the median? To prove your answer is correct, you must exhibit both an algorithm that uses that many comparisons and a proof that there is no faster algorithm. Do the same for 6 numbers.

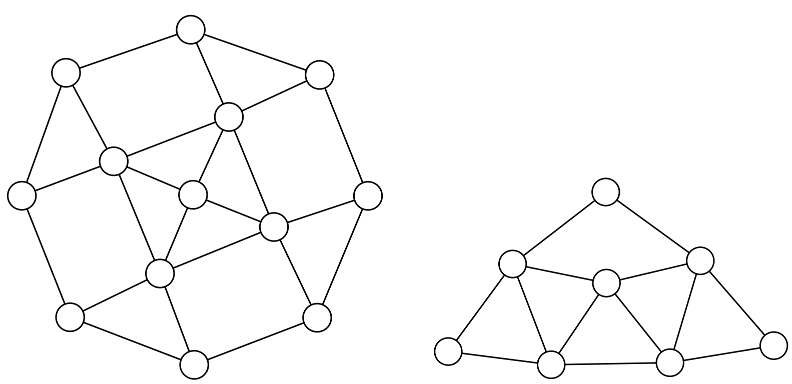
3. PARTITION is the problem of deciding, given a set S of numbers, whether it can be partitioned into two subsets whose sums are equal. (A *partition* of S is a collection of disjoint subsets whose union is S.) SUBSETSUM is the problem of deciding, given a set S of numbers and a target sum t, whether any subset of number in S sum to t.

   (a) Describe a polynomial-time reduction from SUBSETSUM to PARTITION.

   (b) Describe a polynomial-time reduction from PARTITION to SUBSETSUM.

4. Recall from class that the problem of deciding whether a graph can be colored with three colors, so that no edge joins nodes of the same color, is NP-complete.

   (a) Using the gadget in Figure 1(a), prove that 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.]*



(a) (b)

Figure 1. (a) Gadget for planar 3-colorability. (b) Gadget for degree-4 planar 3-colorability.

   (b) Using the previous result and the gadget in figure 1(b), prove that deciding whether a planar graph *with maximum degree 4* 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.]*

5. (a) Prove that if G is an undirected bipartite graph with an odd number of vertices, then G is nonhamiltonian. Describe a polynomial-time algorithm to find a hamiltonian cycle in an undirected bipartite graph, or establish that no such cycle exists.

   (b) Describe a polynomial time algorithm to find a hamiltonian *path* in a directed acyclic graph, or establish that no such path exists.

   (c) Why don't these results imply that P=NP?

6. Consider the following pairs of problems:

   (a) MIN SPANNING TREE and MAX SPANNING TREE

   (b) SHORTEST PATH and LONGEST PATH

   (c) TRAVELING SALESMAN PROBLEM and VACATION TOUR PROBLEM (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? Why?

7. Prove that PRIMALITY (Given n, is n prime?) is in NP $\cap$ co-NP. *[Hint: 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 verified an used to show that n is prime in polynomial time.]*

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