CSEP 590B
Spring Quarter 2006
Assignment 5
Due Thursday May 4, 2006

All solutions should be neatly written or type set. All major steps in proofs and algorithms must be justified.

1. (10 points) In this problem you will explore the connection between "optimization problems" and "decision problems". In an optimization problem, given an input, the output is some object that minimizes or maximizes some quantity. As an example, we have the famous Traveling Salesman Problem (TSP):

Input:	An undirected graph G with weighted edges
Output:	A tour of the graph with total minimim weight. That is, a path through the graph that visits each vertex exactly once and returns to the beginning such that the sum of the weights of the edges on the path is minimized.

   The complexity theory that we are studying is mostly about decision problems (or equivalently languages). A decision problem is one that has a simple yes or no answer which can be defined by a property. The TSP problem above has an corresponding decision problem (DTSP):

Input:	An undirected graph G with non-negative weighted edges, and a number W
Property:	The graph contains a tour with total weight bounded above by W.

   It is clear that TSP is at least as hard as DTSP in the sense that if one has an algorithm for TSP then it can be used to solve the DTSP as follows. Given a graph G and number W, use the algorithm for TSP on the graph G to compute a minumum weight tour. Compute the weight W' of the tour found. If $W' \leq W$ then answer yes to the DTSP problem, else answer no. In some sense DTSP is polynomial time reducible to TSP because the time to solve DTSP given an oracle for TSP is polynomial. Since we know that DTSP is NP-complete, then we say that TSP is NP-hard.

   Consider the following optimization problem called Maximum Cut

Input:	An undirected graph $G = (V, E)$ with non-negative weighted edges
Output:	A set $U \subseteq V$ which maximizes the sum of the weights of the edges from vertices in U to vertices in $V - U$.

   (a) Design a decision problem for Maximum Cut.

   (b) Show that your decision problem is polynomial time reducible to the optimization problem, Maximum Cut.

   (c) Give a polynomial time verifier for the Maximum Cut decision problem.

   (d) It is known that the Maximum Cut decision problem is NP-complete, what conclusion can you make about the optimization problem.

2. (10 points) It is well known that the Subset Sum problem is NP-complete (cf. 268-270 of Sipser). The Subset Sum problem is defined by.

Input:	A sequence of numbers $a_1, a_2, \ldots, a_n$ and a number b all written in binary.
Property:	There is a set $S \subseteq \{1, 2, \ldots, n\}$ such that $\sum_{i \in S} a_i = b$.

   Show that the Equal Partition problem is NP-complete where it is defined by:

Input:	A sequence of numbers $c_1, c_2, \ldots, c_n$ all written in binary.
Property:	There is a set $S \subseteq \{1, 2, \ldots, n\}$ such that $\sum_{i \in S} c_i = \sum_{j \notin S} c_j$.

   Part of your proof should be the construction of a polynomial time reduction of Subset Sum to Equal Partition.