CS 473: Algorithms, Fall 2010
HW 10 (due Thursday, December 2)

This homework contains four problems. **Read the instructions for submitting homework on the course webpage**. In particular, *make sure* that you write the solutions for the problems on separate sheets of paper; the sheets for each problem should be stapled together. Write your name and netid on each sheet.

Collaboration Policy: For this home work, Problems 1-3 can be worked in groups of up to 3 students each.

Problem 0 should be answered in Compass as part of the assessment HW10-Online and should be done individually.

0. (10 pts) HW10-Online on Compass.

1. (25 pts) Let $G = (V, E)$ be an undirected graph. A subset $S \subseteq V$ of nodes in G is called a *covering set* if for all $v \in V$, $v \in S$ or there is some node $u \in S$ such that $\{u, v\} \in E$. In other words every node in $V \setminus S$ is connected by an edge to some node in S. The decision version of the minimum covering set problem is the following: Given a graph G and an integer k, does G have a covering set of size at most k? Prove that this problem is NP-Complete. *Hint:* Use a reduction from Set Cover.

2. (30 pts) Let $G = (V, E)$ be a directed graph that has weights on its edges; $w(e)$ represents the weight of edge e and it can be positive or negative. Given G the Zero-Length-Cycle problem is to check if G has a (simple) cycle C such that the sum of the weights on the edges in C is exactly equal to 0. Show that this problem is NP-Complete.

3. (35 pts) In the Node Disjoint Paths problem, we are given an undirected graph G, k vertices $s_1, s_2, \ldots, s_k$ (the sources), and k vertices $t_1, t_2, \ldots, t_k$ (the destinations). The goal is to decide whether G has k node-disjoint paths (that is, paths which have no nodes in common) such that the i-th path goes from s_i to t_i. Show that the Node Disjoint Paths problem is **NP**-complete. Note that this problem differs from the s-t node-disjoint paths problem in that the paths have to connect different pairs.

   Here is a sequence of progressively stronger hints.
   (a) Reduce from 3-SAT.
   (b) For a 3-SAT formula with m clauses and n variables, use $k = m + n$ sources and destinations. Introduce one source/destination pair (s_x, t_x) for each variable x, and one source/destination pair (s_c, t_c) for each clause c.
   (c) For each 3-SAT clause, introduce 6 new intermediate vertices, one for each literal occurring in that clause and one for its complement.
   (d) Notice that if the path from s_c to t_c goes through some intermediate vertex representing, say, an occurrence of variable x, then no other path can go through that vertex. What vertex would you like the other path to be forced to go through instead?