CSC363 - Summer 2007
Assignment 4

Problem 1 [15p]
Recall that $L \in \mathbf{coNP}$ iff $\overline{L} \in \mathbf{NP}$. Also recall that L is **coNP**-hard iff for all $L' \in \mathbf{coNP}$, $L' \leq_p L$.

1. [5p] Show that if L is **NP**-complete, then $\overline{L}$ is **coNP**-complete.
2. [5p] Show that if $\mathbf{P} = \mathbf{NP}$, then $\mathbf{P} = \mathbf{coNP}$.
3. [5p] Show that if L is **NP**-complete and $L \in \mathbf{coNP}$, then $\mathbf{NP} = \mathbf{coNP}$.

For the rest of this assignment, we will be concerned with some specific graph problems. Some definitions first. You must have seen these in some other course.

In an *undirected* graph G, each edge is a set containing two vertices $\{u, v\}$. One can travel in either direction along this edge. In a *directed* graph G', each edge is an ordered pair of distinct vertices (u, v), meaning that one can travel directly from u to v, but not necessarily backwards. In a directed graph, the two edges (u, v) and (v, u) are distinct and one may be present independent of the other.

A *cycle* in an undirected graph G is a sequence of nodes $(v_1, \ldots, v_k)$ such that for every $i \in \{1, \ldots, k-1\}$, the edge $\{v_i, v_{i+1}\}$ exists in G and $\{v_k, v_1\}$ also exists in G. A cycle in a directed graph G' is also a sequence of the form above, but now the edges must be directed, so for every $i \in \{1, \ldots, k-1\}$, (v_i, v_{i+1}) is an edge in G', and (v_k, v_1) is also an edge in G'. The *size* of a cycle is the number of nodes in it, k in the example above.

A *simple cycle* is a cycle with no repeated vertices or edges. Thus, in an undirected graph the minimum simple cycle has 3 nodes, while in a directed graph the minimum simple cycle has 2 nodes. The reason for this is that, in an undirected graph, going from u to v and backwards, we are repeating an edge. In a directed graph, this is not the case.

In either a directed or an undirected graph, we can assign weights to edges. In the problems below, we will consider integer weights, positive, negative or zero. You may assume that these weights are encoded as usual in binary and are given as part of the description of the graph. The weight of the cycle $C = (v_1, \ldots, v_k)$ is $w_C = w((v_1, v_2)) + \ldots + w((v_{k-1}, v_k)) + w((v_k, v_1))$.

Given a graph with weights on edges, we are interested in determining if it has a simple cycle that has weight 0. We define the following two languages for the undirected and the directed versions of this problem. UZC stands for UNDIRECTED-ZERO-CYCLE and DZC stands for DIRECTED-ZERO-CYCLE:

$$\text{UZC} = \{\langle G \rangle : G \text{ is an } \textit{undirected} \text{ graph with weights on edges that has a simple cycle of weight } 0\}$$

$$\text{DZC} = \{\langle G \rangle : G \text{ is an } \textit{directed} \text{ graph with weights on edges that has a simple cycle of weight } 0\}.$$

Problem 2 [35p]
In this problem, we aim to show that both UZC and DZC are **NP**-complete. For that, we consider the following intermediate problem:

$$\text{ZERO-SUM} = \{\langle S \rangle : S \text{ is a set of integers that has a non-empty subset summing to } 0\}.$$

1. [5p] Argue that both DZC and UZC are in **NP**.
2. [5p] Show that SUBSET-SUM $\leq_p$ ZERO-SUM. You may assume that in the SUBSET-SUM problem, the input set contains only *positive* integers. This should be easy.
3. [10p] Show that ZERO-SUM $\leq_p$ DZC. Hint: add a new edge to the graph for every integer in the set; connect these edges using "neutral" edges; assign directions in order to avoid obtaining a zero-weight cycle of neutral edges.

4. [15p] Show that DZC $\leq_p$ UZC. This is a bit tricky. If you can do it on your own without using the ideas below, do it. Otherwise, I will get you started with an incomplete construction. Your job is to realize why this does not work and how to fix it.

   Given an input G to DZC, construct an input G' to UZC as follows. For every node v of G add exactly two nodes i_v and o_v to G' (they stand for "incoming" and "outgoing"). For every directed edge (u, v) in G, add the undirected edge $\{o_u, i_v\}$ to G' and set $w'(\{o_u, i_v\}) = w((u, v))$. Finally, for every v in G, add the undirected edge $\{i_v, o_v\}$ to G', with weight 0.

   It should be clear that: *if* G has a zero-weight simple cycle, call it $(v_1, \ldots, v_k)$, *then* G' has a zero-weight simple cycle in $(o_{v_1}, i_{v_2}, o_{v_2}, \ldots, o_{v_k}, i_{v_1})$.

   (a) [5p] Show that the other direction is false.

   (b) [2p] Show how to compute a value $B \geq 0$ such that no set of edges from G (cycle or not) can achieve a total weight in absolute value greater than B.

   (c) [8p] Show how to add either $+B$ or $-B$ to the weights of the edges in G' in such a way that the reduction is correct.

Problem 3 [10p]

Consider the search version of UZC, called UZC-SEARCH: given as input an undirected graph G with weights on edges, output either a zero-weight cycle of G, or NIL if no such cycle exists. Show that if a polynomial time algorithm exists for UZC, then we can construct a polynomial time algorithm for UZC-SEARCH.