CS 473: Algorithms, Fall 2009
HW 9 (due Tuesday, November 17 in class)

This homework contains three problems. **Read the instruction for submitting homework on the course web page**. In particular, *make sure* that you write the solutions for the problems on separate sheets of paper. Write your name and netid on each sheet.

Collaboration Policy: For this home work, students can work in groups of up to 3 members each. Each group submits only one written solution (some groups will do an oral presentation. Indicate your group members on the homework (netids are needed)).

1. (30 pts) Problem 7.17 in the text book. As a clarification, you are allowed to do any computation on the original graph (such as finding a maximum flow) but the only way to find which edges have been destroyed is by running the ping tool.

2. (30 pts) Let u, v, w be three distinct nodes in an *undirected* graph G. Describe a linear time algorithm that decides if there is (simple) path in G from u to v that goes via w. Note that a path cannot include a node twice. Give an example of a connected graph G and three vertices u, v, w in G such that there is no simple path from u to v including w. *Hint:* For the algorithm, use network flow with source w.

3. (40 pts) In the 2SAT problem, you are given a set of clauses, where each clause is the disjunction (OR) of two literals (a literal is a Boolean variable or the negation of a Boolean variable). You are looking for a way to assign a value `true` or `false` to each of the variables so that *all* clauses are satisfied—that is, there is at least one true literal in each clause. For example, here's an instance of 2SAT:

$$(x_1 \vee \overline{x}_2) \wedge (\overline{x}_1 \vee \overline{x}_3) \wedge (x_1 \vee x_2) \wedge (\overline{x}_3 \vee x_4) \wedge (\overline{x}_1 \vee x_4) \quad .$$

   This instance has a satisfying assignment: set x_1, x_2, x_3, and x_4 to `true`, `false`, `false`, and `true`, respectively.

   (a) (4 pts) Are there other satisfying truth assignments of this 2SAT formula? If so, find them all.

   (b) (4 pts) Give an instance of 2SAT with four variables, and with no satisfying assignment.

   The purpose of this problem is to lead you to a way of solving 2SAT efficiently by reducing it to the problem of finding the strongly connected components of a directed graph. Given an instance I of 2SAT with n variables and m clauses, construct a directed graph $G_I = (V, E)$ as follows.

   - G_I has $2n$ nodes, one for each variable and its negation.
   - G_I has $2m$ edges: for each clause $(\alpha \vee \beta)$ of I (where α, β are literals), G_I has an edge from the negation of α to β, and one from the negation of β to α.

Note that the clause $(\alpha \vee \beta)$ is equivalent to either of the implications $\overline{\alpha} \Rightarrow \beta$ or $\overline{\beta} \Rightarrow \alpha$. In this sense, G_I records all implications in I.

(c) (4 pts) Carry out this construction for the instance of 2SAT given above, and for the instance you constructed in (b).

(d) (12 pts) Show that if G_I has a strongly connected component containing both x and $\overline{x}$ for some variable x, then I has no satisfying assignment.

(e) (12 pts) Now show the converse of (d): namely, that if none of G_I's strongly connected components contain both a literal and its negation, then the instance I must be satisfiable. (*Hint*: Assign values to the variables as follows: repeatedly pick a sink strongly connected component of G_I. Assign value `true` to all literals in the sink, assign `false` to their negations, and delete all of these. Show that this ends up discovering a satisfying assignment.)

(f) (4 pts) Conclude that there is a linear-time algorithm for solving 2SAT.