HW 9 (due Tuesday, at noon, April 22, 2014)

CS 473: Fundamental Algorithms, Spring 2014 Version: **1.01**

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: The homework can be worked in groups of up to 3 students each.

1. (30 PTS.) Ping.
Problem 7.17 from Kleinberg-Tardos book. We could type up the problem for you but Kleinberg-Tardos book has a very nice chapter on flows and also numerous interesting problems so take this opportunity to look at the book.

2. (40 PTS.) Biconnected graphs and Even Cycles
Let $G = (V, E)$ be an undirected graph. Recall the definition of a cut-vertex from Problem 1 in HW 1. A vertex u is a cut-vertex of G if removing u from G results in at least two disconnected components. A graph G is biconnected (or 2-connected) if G has *no* cut-vertices. Every connected undirected graph G decomposes into a tree of its maximal biconnected components (also called blocks) called the block cut tree. See `http://en.wikipedia.org/wiki/Biconnected_component`. The block cut tree of G can be computed in linear time via DFS following ideas similar to what we saw earlier for finding cut-vertices. In this problem we will understand biconnected components via disjoint paths, maxflow-mincut theorem, and obtain an application to find even-length cycles in a graph. A biconnected graph with two nodes is an edge; we will call a biconnected graph non-trivial if it has at least three nodes.

- What are the blocks of the graph in HW 1, Problem 1.
- Prove that for if $G = (V, E)$ is a non-trivial biconnected graph then for any distinct nodes $s, t \in V$ there are at least two internally node disjoint paths from s to t. In other words there is a cycle in G that contains s and t.
- Prove that for if $G = (V, E)$ is a non-trivial biconnected graph then for any edge $(u, v) \in E$ there is a cycle C containing the edge (u, v).
- Let C be a cycle in an undirected graph G. An edge e is a chord of C if it is not in C and connects two nodes in C. Prove that if C is an odd-length cycle in G that has chord then there is an even-length cycle in G.
- Prove that if G is a biconnected graph then either G is an odd-length cycle C or that G contains an even-length cycle by following the steps below.
  - Let C be any cycle in G. Prove that if there is a node of G not on C then there is an edge (u, v) such that $v \in C$ and $u \notin C$.
  - Prove that there is a path P from u to a node $w \in C$ such that $w \neq v$ and no internal node of P is on C.
- Using the above properties describe a linear time algorithm to determine whether a biconnected graph G has an even-length cycle and output one if it has.

3. (30 PTS.) Algorithm for 2-SAT.
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.

We can solve 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.

(a) 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.

(b) 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*: Consider the meta graph and assign values to the literals in a sink component first and their corresponding negations. Use this process to find a satisfying assignment.)

(c) Deduce a linear-time algorithm for solving 2SAT.