Due at the start of class Tuesday, February 28, 2012.

Problem 1. Let $G = (V, E)$ be a directed graph. The *reversal* of G is a graph $G^R = (V, E^R)$ where the directions of the edges have been reversed (i.e. $E^R = \{(i, j) | (j, i) \in E\}$).

(a) Assuming that G is represented by an adjacency matrix $A[1..n, 1..n]$, give an $O(n^2)$-time algorithm to compute the adjacency matrix A^R for G^R.

(b) Assuming that G is represented by an adjacency list Adj$[1..n]$, give an $O(n + e)$-time algorithm to compute an adjacency list representation of G^R.

Problem 2. Do Exercise 7 on pages 108-109 of Kleinberg and Tardos.

Problem 3. Do Exercise 10 on page 110-111 of Kleinberg and Tardos.

Problem 4. Using depth first search, outline an algorithm that determines whether the edges of a connected, undirected graph can be directed to produce a strongly connected directed graph. If so, the algorithm outputs such an orientation of the edges. (Hint: Show that this can be done if and only if removing an edge leaves a connected graph.)

Problem 5. The *diameter* of a tree $T = (V, E)$ is given by

$$\max_{u,v \in V} \delta(u, v)$$

(where $\delta(u, v)$ is the number of edges on the path from u to v). Describe an efficient algorithm to compute the diameter of a tree, and show the correctness and analyze the running time of your algorithm.

Hint: Use BFS or DFS.