Due Thursday, May 2, 2013.

Problem 1. The *square* of a directed graph $G = (V, E)$ is the graph $G^2 = (V, E^2)$ such that $(x, y) \in E^2$ if and only if for some $z \in V$, both $(x, z) \in E$ and $(z, y) \in E$. That is, G^2 contains an edge between x and y whenever G contains a path with exactly two edges between x and y.

(a) Describe an efficient algorithm for computing G^2 from G using the adjacency matrix representation of G. Analyze its efficiency.

(b) Describe an efficient algorithm for computing G^2 from G using the adjacency list representation of G. Analyze its efficiency.

Problem 2. Assume that the vertices of a graph $G = (V, E)$ are numbered $1, \ldots, V$. A directed (not necessarily simple) cycle of G can be represented by an array of vertices, where each vertex has an edge to the next vertex in the array, and the last vertex has an edge to the first vertex.

Assume you have two cycles in a (directed) graph $G = (V, E)$, represented by two arrays A and B, where the two cyces do not share any edges but do intersect (at at least one node). Describe an efficient algorithm to splice the two cycles together into one cycle represented by an array of nodes C. Give the pseudo code *and* briefly state in English how you algorithm works. (There may be more than one way to splice the two cycles together. Any legitimate splicing is fine.) Analyze its efficiency.

Problem 3. There may be more than one shortest path between two vertices in a weighted graph. Given two specified vertices s and t in a weighted graph $G = (V, E)$, show how to modify Dijkstra's algorithm to find the number of shortest paths from s to t.