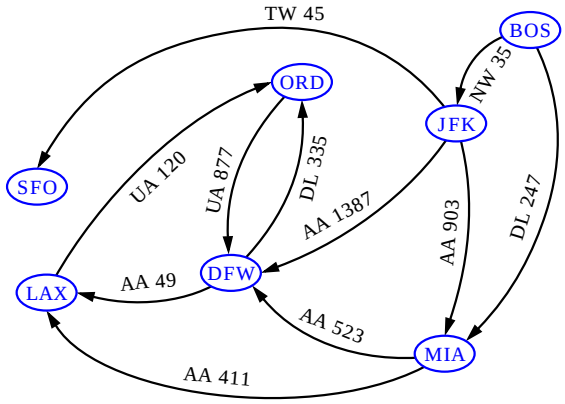
Homework 2 **ICS 163 Graph Algorithms**

Due: Wednesday, January 21, 2003, in class

Please answer the following questions, each of which is worth 10 points.

1. Draw an undirected connected graph that has 18 vertices, two separation edges, three articulation points (separation vertices), and five biconnected components.

2. Show the order of vertices and discovery edges visted in a directed DFS starting from "BOS" for the following graph. Mark every non-tree edge as being a "back edge," "forward edge," or "cross edge."



3. Using the DFS vertex numberings and the vertex states "unvisited," "active," and "done," characterize how we can classify every non-tree edge as being a "back edge," "forward edge," or "cross edge" during a directed DFS.

4. Explain why the DFS traversal runs in $O(n^2)$ time if a graph G is represented with an adjacency matrix, even if the number of edges, m, is much smaller than n^2.

5. Several research labs in the University of ICS are doing a joint project on multimedia. A computer network is built to connect these labs using communication links that form a free tree. They decide to install a file server at one of the labs to share data among all the labs. Since the transmission time on a link is dominated by the link setup and synchronization, the cost of a data transfer is proportional to the number of links used. Hence, it is desirable to choose a "central" location for the file server. Given a free tree T and a node v of T, the *eccentricity* of v is the length of a longest path from v to any other node of T. A node of T with minimum eccentricity is called a *center* of T.

   (a) Design an efficient algorithm that, given an n-node free tree T, computes a center of T.

   (b) Is the center unique? If not, how many distinct centers can a free tree have?