ICS 163 — Algorithms — Winter 2003 — Goodrich — First Midterm

Name:

ID:

1:

2:

3:

4:

5:

total:

1. (50 points). Short Answers.

(a) Define "free tree" in a graph.

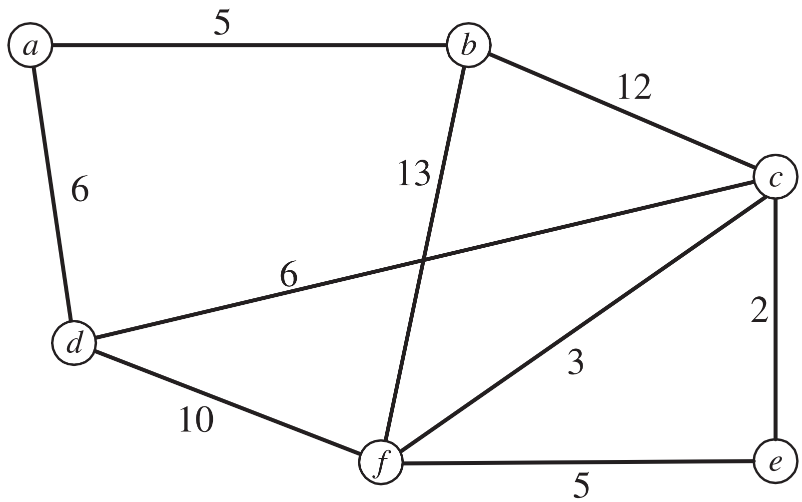
(b) Define "biconnected component."

(c) What is the running time of Dijkstra's algorithm for a connected graph of n vertices and m edges, assuming the graph is represented using an adjacency matrix?

(d) What is the running time of depth-first search on a connected graph with n vertices and m edges that is represented with an adjacency list?

(e) Define "topological ordering" of a directed acyclic graph.

2. (50 points). Consider the following graph:



(a) What is the shortest path distance from a to each of the vertices b, c, d, and e?

(b) List all of the values of the label $D[f]$ that are assigned for the vertex f during a running of Dijkstra's algorithm on the above graph, starting from the vertex a. Note: you need to include ***all*** the different values this label takes during a running of the algorithm.

3. (50 points). Draw a connected graph that has five biconnected components, four articulation vertices, and three separation edges.

4. (50 points). Briefly describe an efficient algorithm for determining if a connected directed graph $G = (V, E)$ is strongly connected. What is the running time of your method, in terms of $n = |V|$ and $m = |E|$?

5. (50 points). Let G be a weighted connected graph that has no negative-weight edges. Define the *distance*, $d(u, v)$, between each pair of vertices u and v in G to be the length of a shortest path connecting u and v. The *diameter* of G is defined as follows:

$$Diameter(G) = \max\{d(u, v), \text{ such that } u \text{ and } v \text{ are in } G\}.$$

Describe how you could use an algorithm described in class to design an efficient algorithm for computing $Diameter(G)$. What is the running time of your algorithm, assuming G has n vertices and m edges?