1. Exercise 22.2-3 (p. 539) (20 Points)
What is the running time of BFS if its input graph is represented by an adjacency matrix and the algorithm is modified to handle this form of input?

2. Problem 22.3-2 (p. 547) (20 Points)
Show how depth-first search works on the following graph (Fig. 22.6 in the text). Assume that the for loop of lines 5-7 of the DFS procedure considers the vertices in alphabetical order, and assume that each adjacency list is ordered alphabetically. Show the discovery and finishing times for each vertex, and show the classification of each edge.

![Figure 1: A directed graph for use in Exercises 22.3-2](image)

3. Exercise 22.3-11 (p. 548) (20 Points)
Show that the Depth-first search of an undirected graph $G$ can be used to identify the connected components of $G$, and that the depth-first forest contains as many trees as $G$ has connected components. More precisely, show how to modify depth-first search so that each vertex $v$ is assigned an integer label $cc[v]$ between 1 and $k$, where $k$ is the number of connected components of $G$, such that $cc[u] = cc[v]$ if and only if $u$ and $v$ are in the same connected component.

4. Problem 22.4-1 (p. 551) (20 Points)
Show the ordering of vertices produced by TOPOLOGICAL SORT when it is run on the dag of Figure 22.8, under the assumption of Exercise 22.3-2.

5. Problem 22.5-2 (p. 557) (20 Points)
Show how the procedure STRONGLY-CONNECTED-COMPONENTS works on the graph of Figure 22.6 in the text. Specifically, show the finishing times computed in line 1 and the forest produced in line 3. Assume that the loop of lines 5-7 of DFS considers vertices in alphabetical order and that the adjacency lists are in alphabetical order.
1. Exercise 22.4-3 (p. 552) (30 Points)

Give an algorithm that determines whether or not a given undirected graph $G = (V, E)$ contains a cycle. Your algorithm should run in $O(V)$ time, independent of $E$.