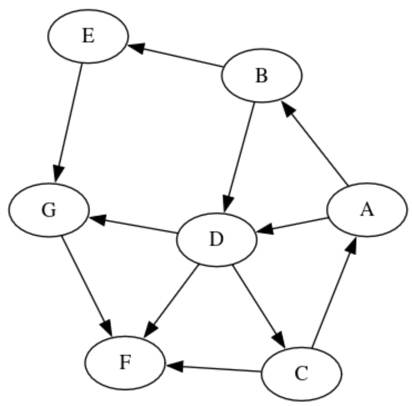
The exam will be in class (one hour) and closed book. You may use one page of notes (one side of an 8.5×11" piece of paper, hard copy required), which will be handed in with the exam.

It is strongly recommended that you treat this as a practice exam and work on it in a similar environment.

---

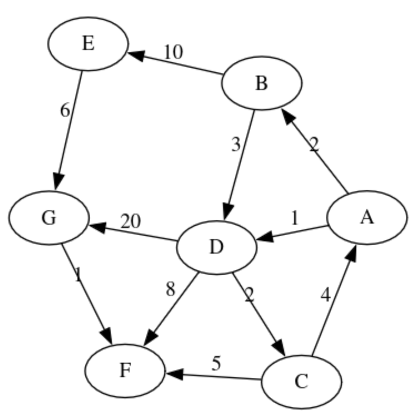
1. Carry out depth-first search on the graph shown below. Use the recursive DFS discussed in class, not a stack-based implementation. Start from vertex A. Use alphabetical order to break ties.

   Indicate the order in which vertices are marked as "discovered" and as "processed". Showing more of your work, such as a tree of the recursive calls, can help you earn partial credit in case you make a mistake.



2. Carry out Dijkstra's algorithm on the graph shown below. Start from vertex A. Use alphabetical order to break ties.

   Give the final `dist` labels for each vertex. Also show your work by indicating the order in which vertices are removed from the priority queue and showing a table of the `dist` labels as they are updated in each iteration.



3. For each of the following scenarios, explain how to solve the problem by modeling it as a graph problem — address the points below.

   - A public health organization needs to set up emergency medical supply routes between a group of remote villages in a mountainous region. Each potential path between two villages takes a different amount of time and effort to maintain due to terrain and distance. The organization hopes to ensure that every village is reachable from any other by some route but has a limited budget. Given the organization's budget and the cost of each potential path between two villages, determine whether it is possible to connect all of the villages and, if so, how to do it.
   - After a storm knocks out some roads, how many separate groups of towns are left where each town is reachable from the others?
   - Software packages may depend on components provided by other packages which must be installed first. For a collection of packages, are there circular dependencies that would prevent installation?
   - In a communications network, are there any critical elements — a server or a cable — which would split the network if they failed or were removed?
   - In a city's public transportation network, the cost of a route depends on the number of transfers rather than the distance traveled. Find routes with the minimum number of transfers.
   - You are organizing a conference with a number of sessions. When registering for the conference, participants were asked for up to two sessions that they most wanted to attend. Sessions can be scheduled in either the morning or the afternoon, and participants can only attend one session in each time slot. Determine how to schedule the sessions so that every participant is able to attend their preferred sessions, or determine that such a schedule is not possible.

   (a) Graph representation.

   - What do the vertices of the graph represent?
   - What do the edges of the graph represent?
   - Is the graph undirected or directed? If directed, what does the direction of the edges reflect?
   - Is the graph weighted or unweighted? If weighted, what are the weights?
   - What is the solution to the problem in terms of the graph?

   (b) Properties of the graph. Provide a brief explanation with each answer.

   - Is the graph simple or not simple?
   - Is the graph sparse or dense?
   - Is the graph cyclic or acyclic?

(c) Algorithm. Give an efficient algorithm for solving the problem in terms of the well-known graph algorithms discussed in class. Design graphs, not algorithms — treat the well-known graph algorithms as black boxes rather than something to modify and design your graphs so that you can use them as-is. You do not need to describe the algorithm itself, just identify it and explain how it is used to obtain the answer to the original problem.

4. For each of the following scenarios, address the tradeoffs with using an adjacency list vs an adjacency matrix representation for the graph. Is there a clear winner? Does it not matter? Consider both time and space requirements, and give a brief explanation for your answer.

   (a) The graph has 10 vertices and 20 edges, and you need to frequently use both `areAdjacent` and `adjacentVertices`.

   (b) The graph has 1,000 vertices and 100,000 edges, and you need to frequently use `areAdjacent`.

   (c) The graph has 1,000 vertices and 800,000 edges, and you need to frequently use `adjacentVertices`.