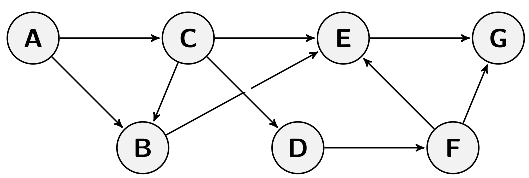
Collaboration Policy: You are encouraged to collaborate with up to 4 other students, but all work submitted must be your own *independently* written solution. List the computing ids of all of your collaborators in the `collabs` command at the top of the tex file. Do not share written notes, documents (including Google docs, Overleaf docs, discussion notes, PDFs), or code. Do not seek published or online solutions for any assignments. If you use any published or online resources (which may not include solutions) when completing this assignment, be sure to cite them. Do not submit a solution that you are unable to explain orally to a member of the course staff. Any solutions that share similar text/code will be considered in breach of this policy. Please refer to the syllabus for a complete description of the collaboration policy.

Collaborators: list your collaborators

Sources: list your sources

PROBLEM 1 *Depth First Search Returns*

Recall the graph from PS1's DEPTH FIRST SEARCH problem and your solution, after performing a *dfs_rec()* starting at vertex *A* on this graph. *Remember that when there was an equally valid choice of which vertex to visit next, you chose based on increasing alphabetical order (e.g., if B and C are both valid choices for the next visit, choose B first).*



1. Using your seen/discovery and done/finish times from PS1, give the specific *Topological Ordering* that would be produced by the DFS-based algorithm we studied in class. If there is no valid topological sorting for this graph, write "None".

   Solution:

2. List all strongly connected components, or write "None" if there are none.

   Solution:

3. What **one** edge could be added to create one strongly connected component containing all nodes in the graph? Write "None" if there isn't one.

   Solution:

PROBLEM 2 *Distance from Destination*

You are part of logistics planning for a major event at UVA. There are several locations that you could host your event, each near a UTS bus stop. Your primary concern is minimizing the amount of time that it takes for a bus passenger to get to the event. You have access to the UTS bus schedule that contains all bus routes and the time to get from one stop to the next. You read this information as a directed graph, with each bus stop as a node, directed edges between stops indicating the bus directions, and edge weights representing the time it takes a bus to get from one stop to the next stop.

Create an algorithm that can identify the most central bus stop; that is, the stop such that the longest path to reach it—in terms of time on the bus—from all other stops is minimized. You may ignore time spent waiting to transfer busses or waiting to catch the bus. Describe your algorithm and analyze its runtime.

Solution:

PROBLEM 3 *Ancient Population Study*

Historians are studying the population of the ancient civilization of *Algorithmica*. Unfortunately, they have only uncovered incomplete information about the people who lived there during Algorithmica's most important century. While they do not have the exact year of birth or year of death for these people, they have a large number of possible facts from ancient records that say when a person lived relative to when another person lived.

These possible facts fall into two forms:

- The first states that one person died before the another person was born.
- The second states that their life spans overlapped, at least partially.

The Algorithmica historians need your help to answer the following questions. First, is the large collection of uncovered possible facts internally consistent? This means that a set of people could have lived with birth and death years that are consistent with all the possible facts they've uncovered. (The ancient records *may not be accurate*, meaning all the facts taken together cannot possibly be true.) Second, if the facts are consistent, find a sequence of birth and death years for all the people in the set such that all the facts simultaneously hold. (Examples are given below.)

We'll denote the n people as $P_1, P_2, \ldots, P_n$. For each person P_i, their birth-year will be b_i and their death-year will be d_i. (Again, for this problem we do not know and cannot find the exact numeric year value for these.)

The possible facts (input) for this problem will be a list of relationships between two people, in one of two forms:

- P_i *prec* P_j (indicates P_i died before P_j was born)
- P_i *overlaps* P_j (indicates their life spans overlapped)

If this list of possible facts is not consistent, your algorithm will return "not consistent". Otherwise, it will return a possible sequence of birth and death years that is consistent with these facts.

Here are some examples:

- The following facts about $n = 3$ people are **not** consistent: P_1 *prec* P_2, P_2 *prec* P_3, and P_3 *prec* P_1.
- The following facts about $n = 3$ people **are** consistent: P_1 *prec* P_2 and P_2 *overlaps* P_3. Here are two possible sequences of birth and death years:
  $b_1, d_1, b_2, b_3, d_2, d_3$
  $b_1, d_1, b_3, b_2, d_2, d_3$
  (Your solution only needs to find one of any of the possible sequences.)

Your answer should include the following. Clearly and precisely explain the graph you'll create to solve this problem, including what the nodes and edges will be in the graph. Explain how you'll use one or more of the algorithms we've studied to solve this graph problem, and explain why this leads to a correct answer. Finally, give the time-complexity of your solution.

Solution: