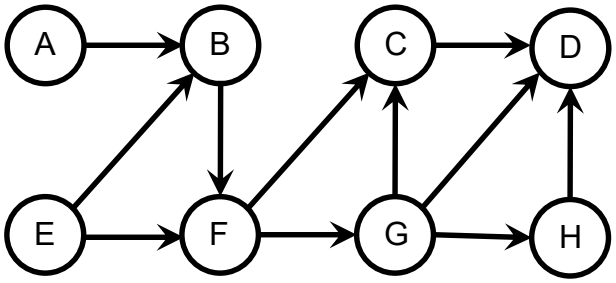
1. [8 points] Find a topological ordering of the nodes in the digraph below, or explain why there is no such ordering. You must show work, corresponding to an appropriate algorithm, if you want full credit.



Answer:

I'm going to use an iterated depth-first traversal, choosing the next node based on alphabetical order if there are multiple candidates, starting the first traversal at A:

The traversals go like this:

First traversal:

```
A --> B --> F --> C --> D                D
                  C <--                  C  D
            F <--                        F  C  D
            F --> G --> H                H  C  D
                  G <--                  G  H  C  D
            F <--                        F  G  H  C  D
      B <--                              B  F  G  H  C  D
A <--                                    A  B  F  G  H  C  D
```

Second traversal:

```
E                                        E  A  B  F  G  H  C  D
```

Other valid orderings are possible:

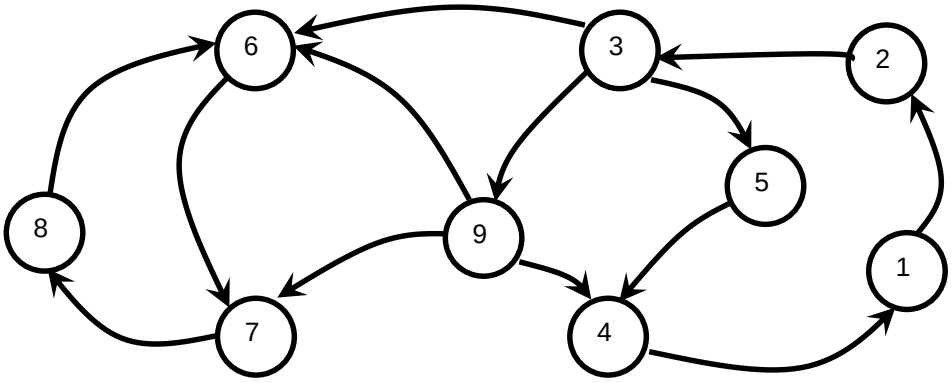
Either A or E could be first, but the first two must be A and E.

B, F, and G must come next, in that order, and w/o any other tasks inserted.

C and H must precede D, but they could be in either order.

IOW: {E A} B F G {H C} D

For Questions 2 - 6, consider the following digraph:



When performing traversals on the graph, if there is more than one unvisited successor to consider, process those successors in the order of their labels.

2. **[8 points]** If a depth-first traversal was performed on the graph shown above, starting at node 1, which nodes would be visited, and in what order, before the first backtracking step?

   Answer:

   A depth-first traversal, starting at node 1, would go as follows:

   1 --> 2 --> 3 --> 5 --> 4

   At this point, node 4 has no unvisited successors, so we must backtrack.

3. **[4 points]** Continuing from question 2, at what node would the first backtracking end?

   Answer:

   From 4, the traversal would backtrack to 5 (still no unvisited successors), and then to 3 (which has two unvisited successors, 6 and 9).

4. **[4 points]** Is the graph shown above strongly connected? Justify your conclusion.

   Answer:

   To be strongly connected, given any two nodes X and Y, there must be a directed path in the graph from X to Y.

   In this graph, there is no directed path from 6 or 7 or 8 to any other nodes. There are other examples where no path exists.

5. [4 points] Is there a topological ordering of the node in the digraph shown above? Justify your conclusion.

Answer:

A digraph has a topological ordering unless the graph contains a directed cycle.

This graph contains a directed cycle involving nodes 1, 2, 3, 5, and 4.

It also contains other directed cycles. Other valid arguments include:

- **every node has an outgoing edge, implying that every node is the predecessor of some other node, implying that no node could come last in a linear ordering**
- **every node has an incoming edge, implying that every node is the successor of some other node, implying that no node could come first in a linear ordering**

6. [8 points] If a breadth-first traversal is performed on the digraph shown above, starting at node 9, in what order will the nodes be visited?

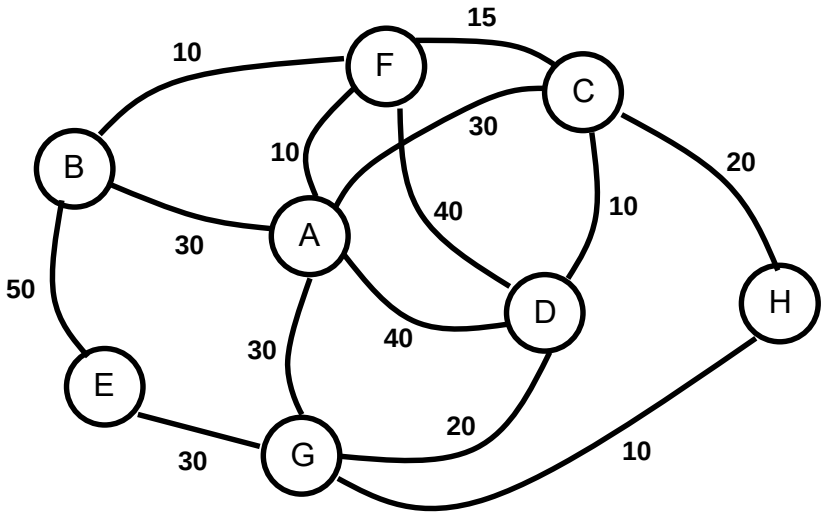
Answer:

scheduling queue	pop/visit
9	initialize with start node
4 6 7	9
6 7 1	4
7 1	6
1 8	7
8 2	1
2	8
3	2
5	3
	5

7. [8 points] Apply Dijkstra's SSAD algorithm to the graph shown below, using A as the start node. Write your final answer as a table; for each node in the graph, show the length of the shortest path from A to that node, and the corresponding path, like so:

```
Node    Distance        Path
-------------------------------------
  R         73          A L N H E R
```

In the first column of the table, list the nodes in alphabetical order. Show work if you want full credit.



Answer:

Here's my analysis; when a node is added to the solved set, I'm recording that node when I update th4 distance for any of its neighbors (so I can reconstruct the paths easily):

```
A     B     C     D     E     F     G     H
-------------------------------------------
0    30,A  30,A  40,A inf    10,A  30,A inf
     20,F  25,F  40    inf     +    30    inf
      +    25    40     70,B        30    inf
            +    35,C  70          30     45,C
                 35    60,G         +     40,G
                  +    60                 40
                       60                  +
                          +
```

And, here is the final result:

```
node   distance    path
-----------------------------------------
A            0
B           20    A  F  B
C           25    A  F  C
D           35    A  F  C  D
E           60    A  G  E
F           10    A  F
G           30    A  G
H           40    A  G  H
```

Note: work was explicitly required and so was Dijkstr's SSAD algorithm, so you lost points if your answer did not include work that actually illustrated use of the specified algorithm.

8. **[6 points]** Suppose we are performing searches on a huge collection of records, stored in a file on disk. In order to improve performance, we have decided to use a buffer pool, with LRU replacement, storing up to 5 records.

The last few searches targeted records at the offsets shown below (in the order shown):

974, 325, 764, 812, 974, 325, 764, 764, 614, 325, 614, 325, 764, 883, 325

List the offsets of records that will be in the buffer pool after the last search, in MRU to LRU order.

Answer:

The answer is easily obtained if you notice that it will be the last five distinct references in the list given above.

MRU				LRU
325	883	764	614	974