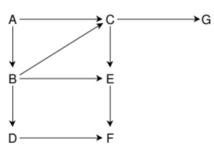
3500 hw 3

Casey Provitera

February 2024

1 Question 1 (20 points)

Perform depth-first search on the given graph, beginning from vertex A. If there are multiple vertex options, choose the one that comes first alphabetically.



(a) original

(b) re-writen

(a) Complete the table with the discovery and finish times for each vertex.

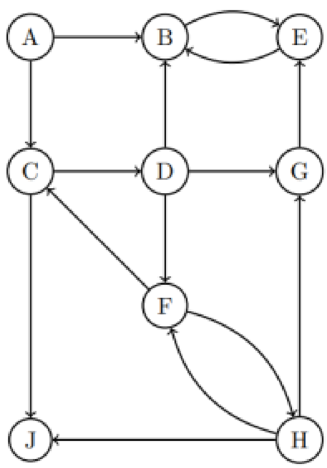
	A	B	C	D	E	F	G
Discovery	1	2	3	11	4	5	8
Finish	14	13	10	12	7	6	9

(b) Next, Assign a label (tree (T), back (B), forward (F), or cross (C) edge) to each edge in the provided graph. Column is edge head.

	A	B	C	D	E	F	G
A	-						
B	T	-					
C	F	T	-				
D		T		-			
E		F	T		-		
F				C	T	-	
G			T				-

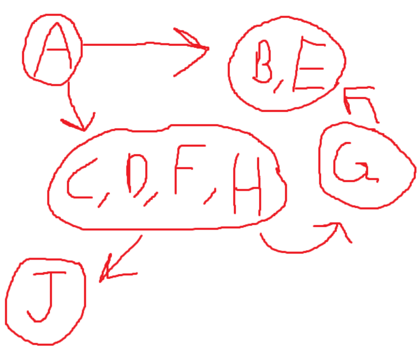
2 Question 2 (30 points)

For the directed graph below



(a) find the strongly connected components.
The strongly connected components are as follows :
(B,E), (C,D,F,H), (A), (J), (G)

(b) draw the DAG of strongly connected components.



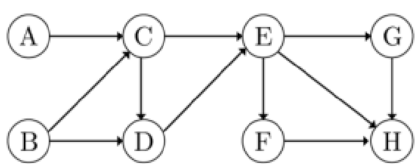
(c) Provide a valid topological sort for the directed acyclic graph (DAG) that you created in part b.
Topographical sort is sorting so all edges go in the same direction, must start with a vertex that has no "in" edges. After removing the current vertex with no "in" edges there must exist another vertex

with no "in" edges. Below is a valid topological sort. Parenthesis show groups.

A, (C, D, F, H), J, G, (B, E)

3 Question 3 (20 points)

Consider the following graph, where each node represents a course. Find an order in which these courses can be taken, ensuring none are taken without meeting the necessary requirements.
Need to take prerequisites classes first, since A and B have an in degree of 0 either can be taken first, I chose to do A first. This then reduces the in degree of C to 0 ect. This is how the below solution is formed. Once vertex E is removed either of F or G can be used I chose to do F first.
A - B - C - D - E - F - G - H



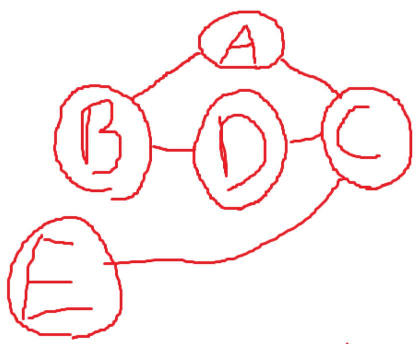
4 Question 4 (30 points)

State whether the following statements are true or false, and explain why:

1. A depth-first search of a directed graph always produces the same number of tree edges (i.e. independent of the order in which the vertices are provided and independent of the order of the adjacency lists)
   The above statement is False. Take the given graph in Q3 as an example, if a depth fist search is performed on C the the resulting output contains 6 edges, but if a depth first search is performed on A the resulting output

has 1 more edge which is the edge from A->C. This means that a depth first search can be performed on the same graph but starting at different vertices and can give a different number of edges in the output.

2. Given an undirected, unweighted, connected graph G, suppose we run a DFS on G starting on some node s. We find that the DFS tree has the property that for any vertex v, the path in the DFS tree from s to v is the shortest (fewest number of edges) path from s to v in G. Then G is a tree.



The above graph is an undirected unweighted graph. If depth first search is run on the above graph from A visiting B first then the distance from A to E will be 4, but the shorted path from A to E is 2. This happens because there is a cycle of (A, B, C, D). For Depth first search to always find the shortest path from any vertex to any other vertex in a graph (A) then there can not be any cycles in A. When there are no cycles in a graph it is a tree proving statement 2.