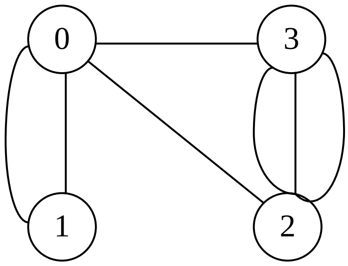
EE2410 Data Structure Hw #5 (Chapter 6 Graphs)
due date 6/2/2024 (Sun.), 23:59
Student No.: ____111060005_____
Name: _______胡昱煊_______

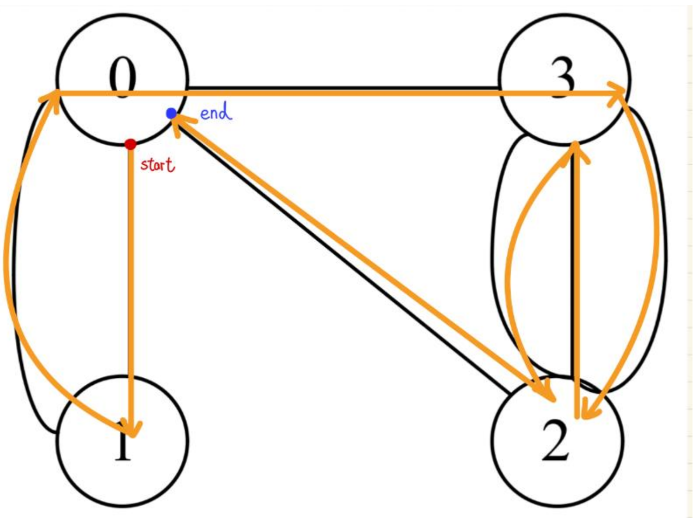
Format: Use MS Word to **edit this file** by directly typing your student number and name in above blanks and your answer to each homework problem right in the **Sol:** blanks as shown below. Then save your file as **Hw5-SNo.pdf**, where SNo is your student number. Submit the **Hw5-SNo.pdf** file via eLearn. The grading will be based on the correctness of your answers to the problems, and the **format requirement**. Fail to comply with the aforementioned format (file name, header, problem, answer, problem, answer,…), will certainly degrade your score. If you have any questions, please feel free to ask. Submit your homework before the deadline (midnight of 6/13 Sun.). Fail to comply (**late** homework) will have ZERO score. **Copy** homework will have ZERO score (both parties) and SERIOUS consequences.

Graphs:

1. (4%) Does the multigraph below have an Eulerian walk? If so, find one.



Sol:

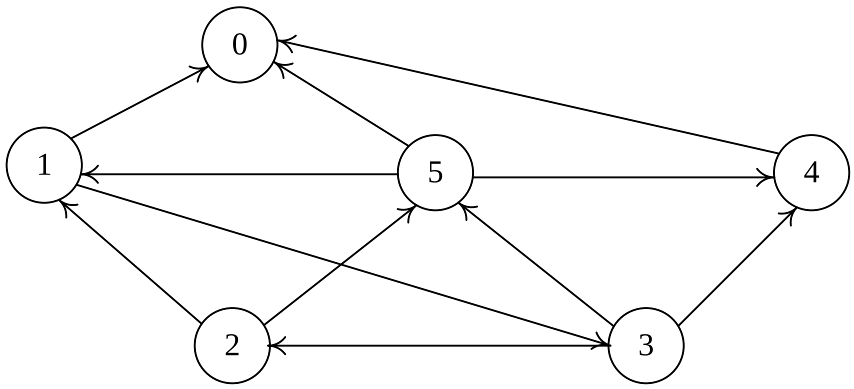


2. (16%) For the digraph below obtain
   (a) The in-degree and out-degree of each vertex

(b) Its adjacency-matrix

(c) Its adjacency-list representation

(d) Its strongly connected components



Sol:

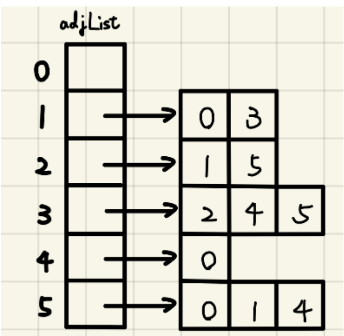
(a)

Node	in-degree	out-degree
0	3	0
1	2	2
2	1	2
3	1	3
4	2	1
5	2	3

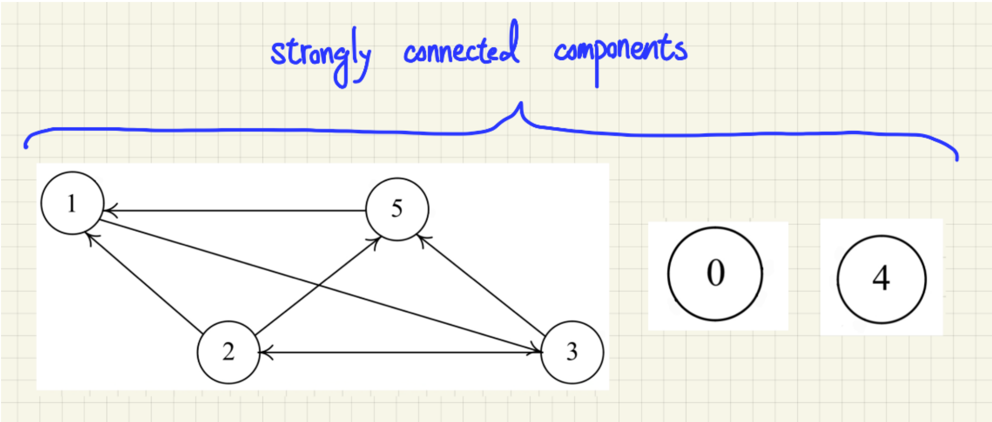
(b)

$$\begin{array}{c c} & \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

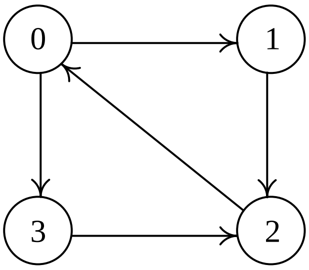
(c)



(d)



3. (4%) Is the digraph below strongly connected? List all the simple paths.



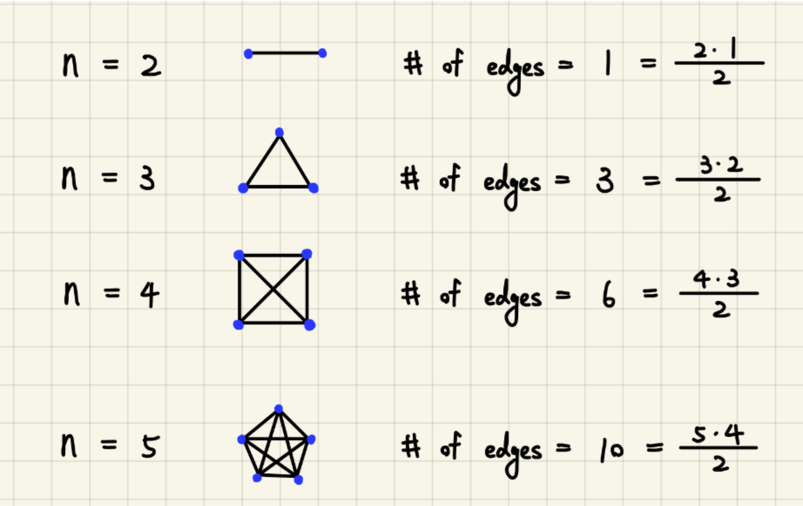
Sol:

Yes, the digraph is strongly connected.

Starting vertex	Path to travel every vertex
0	$0 \to 1 \to 2 \to 0$ $0 \to 1$ $0 \to 1 \to 2$ $0 \to 3$
1	$1 \to 2 \to 0$ $1 \to 2 \to 0 \to 1$ $1 \to 2$ $1 \to 2 \to 0 \to 3$
2	$2 \to 0$ $2 \to 0 \to 1$ $2 \to 0 \to 1 \to 2$ $2 \to 0 \to 3$
3	$3 \to 2 \to 0$ $3 \to 2 \to 0 \to 1$ $3 \to 2$ $3 \to 2 \to 0 \to 3$

4. (6%) Draw the complete undirected graphs on two, three, four, and five vertices. Prove that the number of edges in an n-vertex complete graph is n(n-1)/2.

Sol:



<proof>

Select one vertex, since each edge can be formed by connecting the vertex to the other vertices, the total count will be $n(n-1)$. However, we need to divide the count by 2 since each edge will be counted twice by the method above. Therefore, the number of edges in an undirected complete graph is $\frac{n(n-1)}{2}$.

5. (4%) Apply depth-first and breadth-first searches to the complete graph on four vertices. Assume that vertices are numbered 0 to 3, are stored in increasing order in each list in the adjacency-list representation, and both traversals begin at vertex 0. List the vertices in the order they would be visited.

Sol:

DFS	$0 \to 1 \to 2 \to 3$
BFS	$0 \to 1 \to 2 \to 3$

6. (6%) Let *G* be a graph whose vertices are the integers 1 through 8, and let the adjacent vertices of each vertex be given by the table below:

Vertex	***Adjacent Vertices***
1	(2, 3, 4)
2	(1, 3, 4)
3	(1, 2, 4)
4	(1, 2, 3, 6)

5	(6, 7, 8)
6	(4, 5, 7)
7	(5, 6, 8)
8	(5, 7)

Assume that, in a traversal of *G*, the adjacent vertices of a given vertex are returned in the same order as they are listed in the table above.

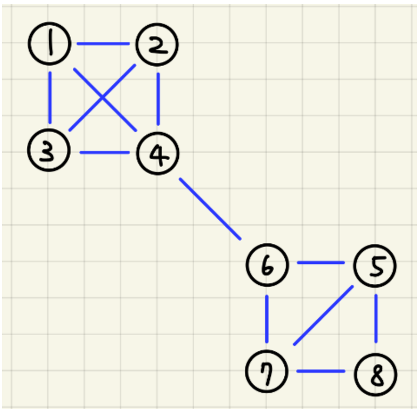
(a) Draw *G*.

(b) Give the sequence of vertices of *G* visited using a DFS traversal starting at vertex 1.

(c) Give the sequence of vertices visited using a BFS traversal starting at vertex 1.

Sol:

(a)



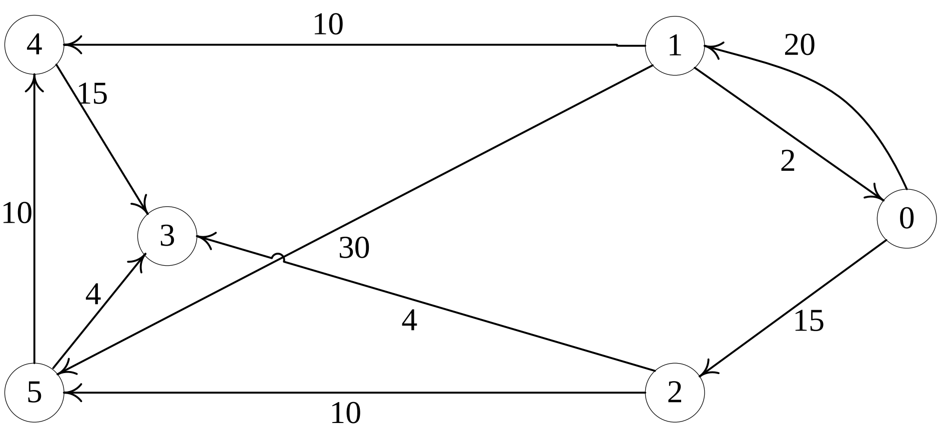
(b)

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 6 \rightarrow 5 \rightarrow 7 \rightarrow 8$$

(c)

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 6 \rightarrow 5 \rightarrow 7 \rightarrow 8$$

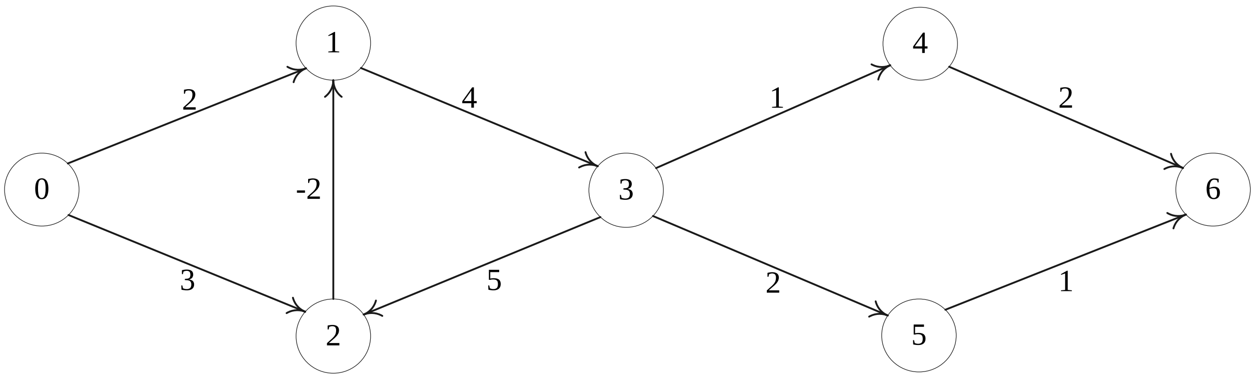
7. (10%) Use ShortestPath (Program 6.8) (Dijkstra's algorithm) to obtain, in nondecreasing order, the lengths and the paths of the shortest paths from vertex 0 to all remaining vertices in the graph below.



Sol:

	path	length
1	0 → 2	15
2	0 → 2 → 3	19
3	0 → 1	20
4	0 → 2 → 5	25
5	0 → 1 → 4	30

8. (10%) Using the directed graph below, explain why ShortestPath (Program 6.8) will not work properly. What is the shortest path between vertices 0 and 6?



sol:

	0	1	2	3	4	5	6
Dist[i]	0	2	3	∞	∞	∞	∞
P[i]	-	0	0	-	-	-	-

↓

	0	1	2	3	4	5	6
Dist[i]	0	2	3	6	∞	∞	∞
P[i]	-	0	0	1	-	-	-

↓

	0	1	2	3	4	5	6
Dist[i]	0	1	3	6	∞	∞	∞
P[i]	-	2	0	1	-	-	-

↓

	0	1	2	3	4	5	6
Dist[i]	0	1	3	6	∞	∞	∞
P[i]	-	2	0	1	-	-	-

↓

	0	1	2	3	4	5	6

Dist[i]	0	1	3	6	7	8	∞
P[i]	-	2	0	1	3	3	-

↓

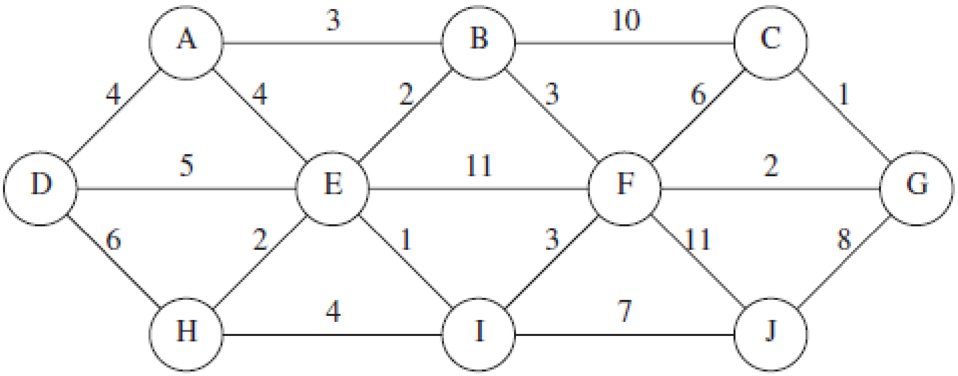
	0	1	2	3	4	5	6
Dist[i]	0	1	3	6	7	8	9
P[i]	-	2	0	1	3	3	4

Discussion of the shortest path between vertex 0 and 6:

Path	path length
0 → 2 → 1 → 3 → 4 → 6	8
0 → 1 → 3 → 4 → 6	9

Since Dijkstra's algorithm cannot process negative-weighted graph, the program will not work properly.

9. (10%) For the weighted graph G shown below,



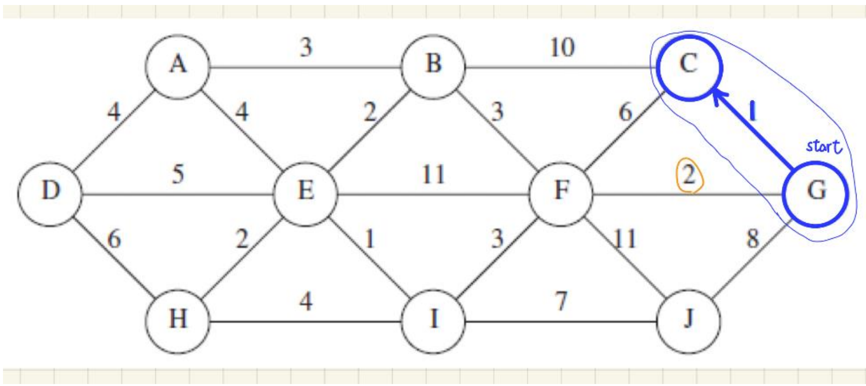
(a) Find a minimum spanning tree for the graph using both Prim's and Kruskal's algorithms.

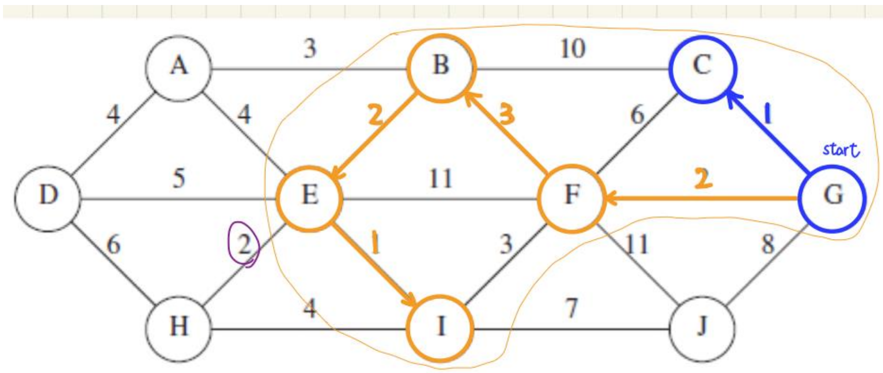
(b) Is this minimum spanning tree unique? Why?

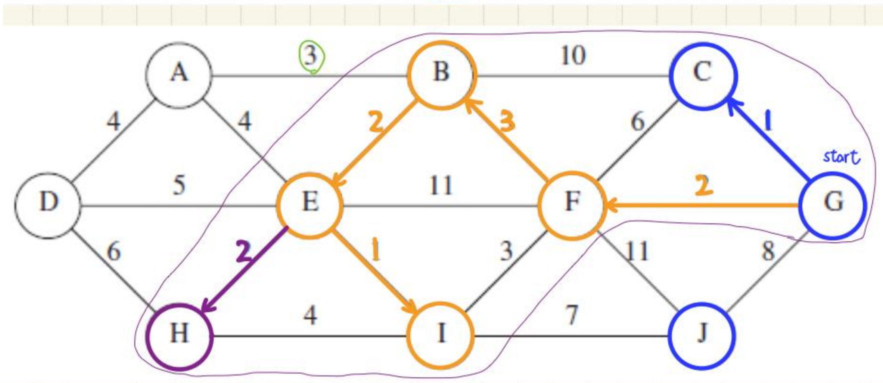
Sol:

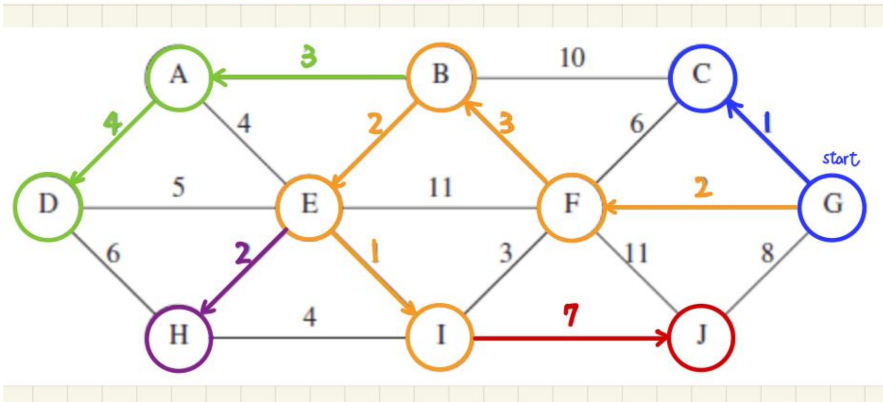
(a)

By Prim's algorithm:

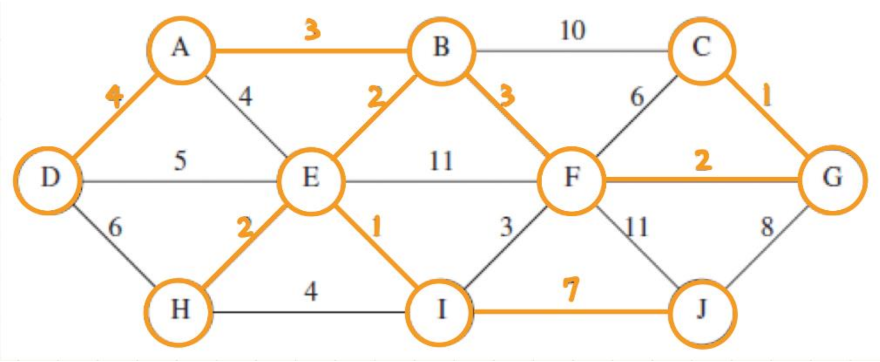






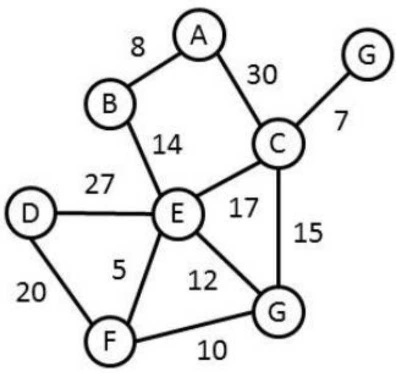


By Kruskal’s algorithm:



(b)
No, since edge(F,B) and edge(F,I) have the same weight of 3, which can form different minimum spanning tree, the minimum spanning tree is not unique.

10. (10%) Answer the questions using the following graph.



(a) Draw the sequence of edges (represented by edge weight) added to the minimal-cost spanning tree generated by the Kruskal's algorithm.

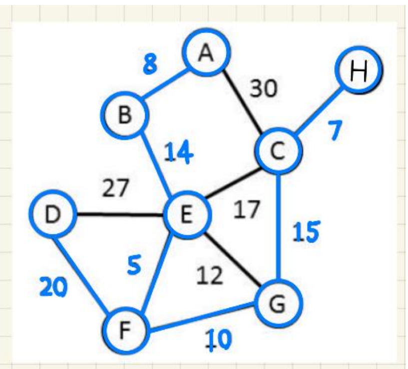
(b) Do the same thing as (a) using Prim's algorithm.

(c) Do the same thing as (a) using Sollin's algorithm.

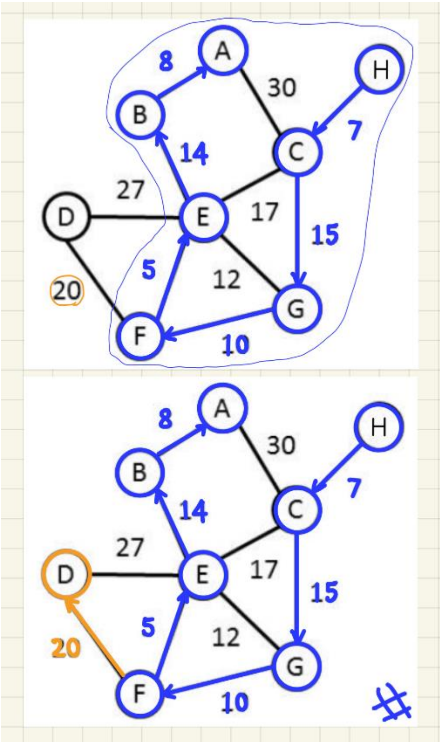
(d) Find the shortest paths from E to all other vertices using Dijkstra's algorithm.

Sol:

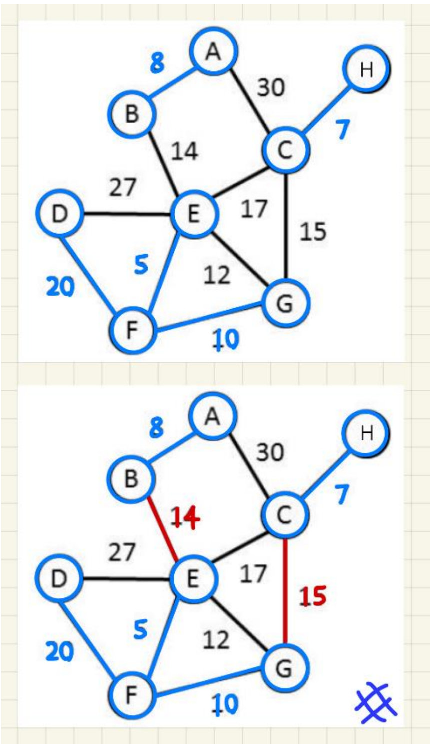
(a)



(b)



(c)



(d)

	path	length

1	$E \to F$	5
2	$E \to G$	12
3	$E \to B$	14
4	$E \to C$	17
5	$E \to B \to A$	22
6	$E \to C \to H$	24
7	$E \to F \to D$	25

11. (4%) Does the following set of precedence relations (<) define a partial order on the elements 0 through 4? Why?

   $0 < 1;\ 1 < 3;\ 1 < 2;\ 2 < 3;\ 2 < 4;\ 4 < 0$

Sol:

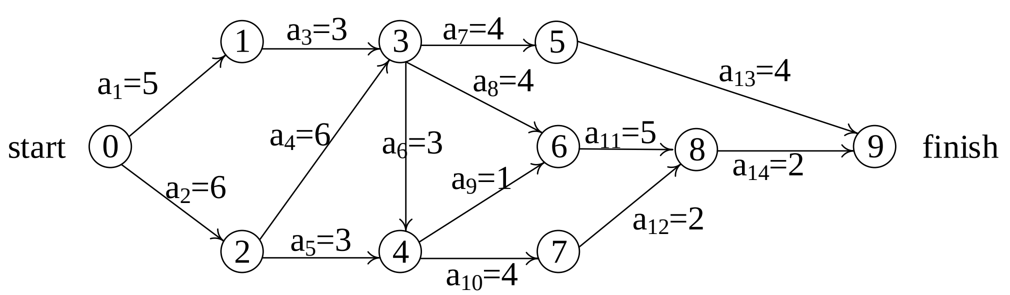
No.

$0 < 1$ and $1 < 2$ implies $0 < 2$;

$0 < 2$ and $2 < 4$ implies $0 < 4$;

$0 < 4$ and $4 < 0$ implies $4 < 4$, which violates irreflexivity of a partial order.

12. (16%) For the AOE network shown below,
   (a) Obtain the early, $e(a_i)$, and late, $l(a_i)$, start times for each activity. Use the forward-backward approach.
   (b) What is the earliest time the project can finish?
   (c) Which activities are critical? Fill the table below for answers to (a), (b), and (c).
   (d) Is there any single activity whose speed-up would result in a reduction of the project finish time?



Sol:

(a) (c)

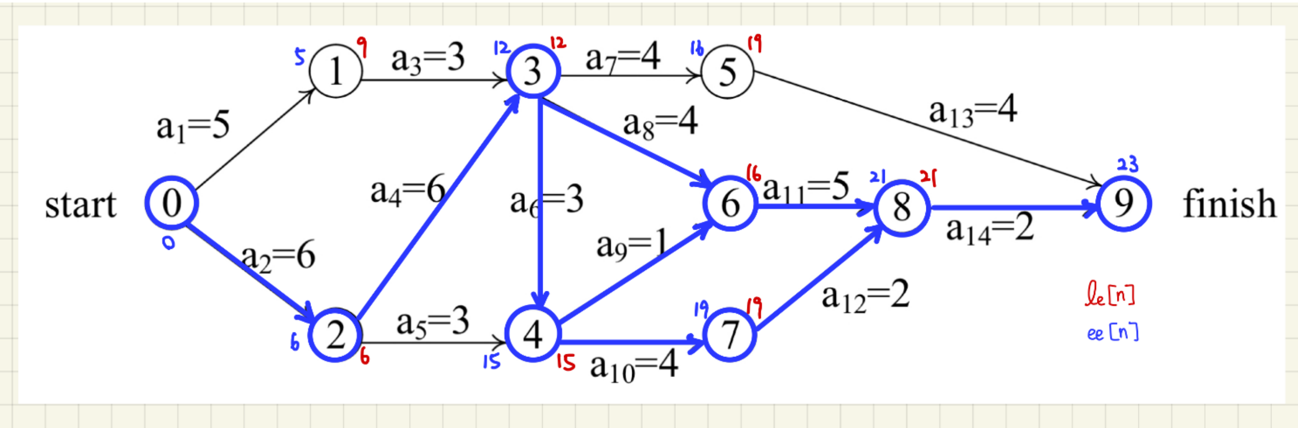
activity	Early time	Late time	slack	critical
	$e(a_i)$	$l(a_i)$	$l(a_i)- e(a_i)$	
a_1	0	4	4	False

a_2	0	0	0	True
a_3	5	9	4	False
a_4	6	6	0	True
a_5	6	12	6	False
a_6	12	12	0	True
a_7	12	15	3	False
a_8	12	12	0	True
a_9	15	15	0	True
a_{10}	15	15	0	True
a_{11}	16	16	0	True
a_{12}	19	19	0	True
a_{13}	16	19	3	True
a_{14}	21	21	0	True

(b) Earliest time = 23.

(d)

Critical path (marked in blue):



a_2, a_4, a_{14} are critical paths that do not have alternative path. Speeding up any of them may reduce the project finish time.