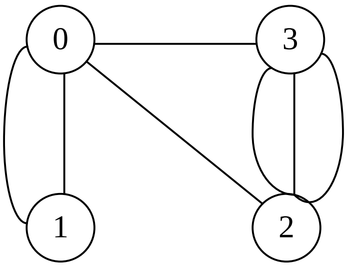
EECS2040 Data Structure Hw #5 (Chapter 6 Graph)
due date 5/30/2022
by 108011235 陳昭維

(Part 1: 2% of final Grade)

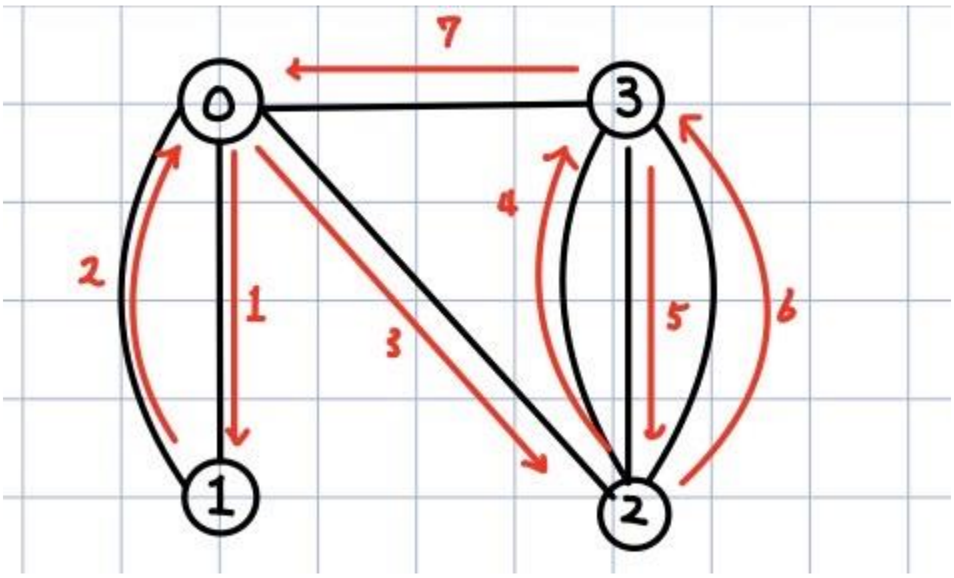
Part 1

1. (10%) Does the multigraph below have an Eulerian walk? If so, find one.

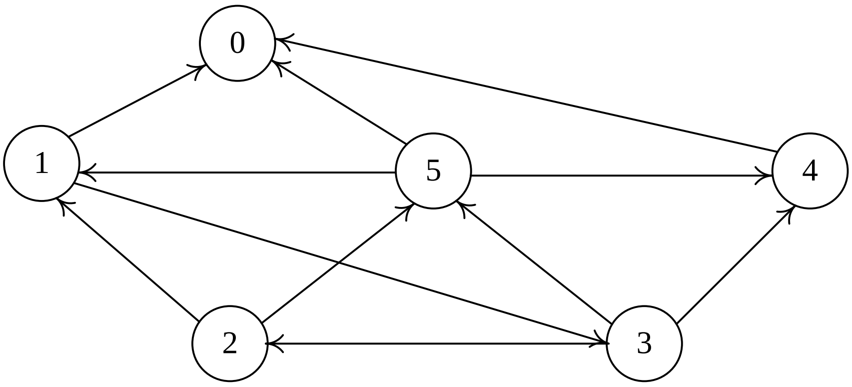


<answer>

Yes there is an Eulerian walk



2. (10%) 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



<answer>

(a)

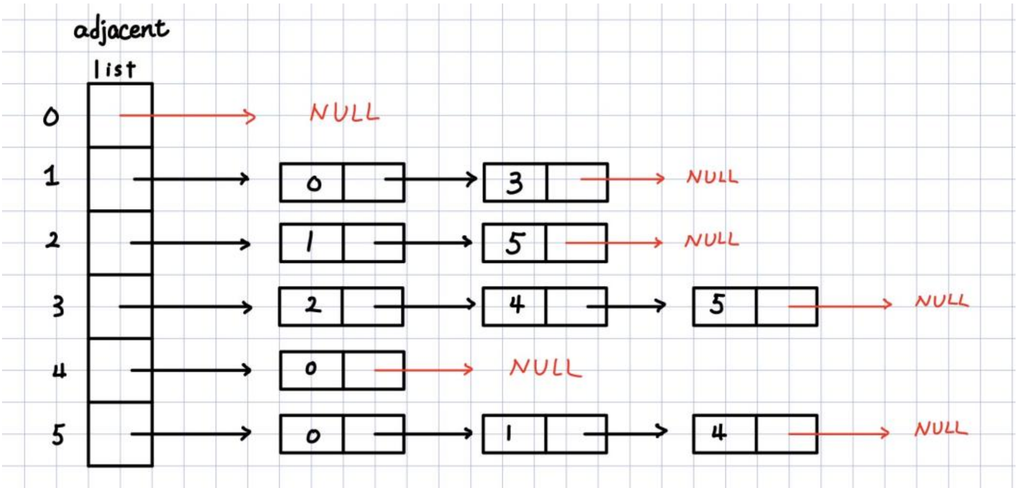
Vertex	In-degree	Out-degree
0	3	0
1	2	2
2	1	2
3	1	3
4	2	1
5	2	3

(b)

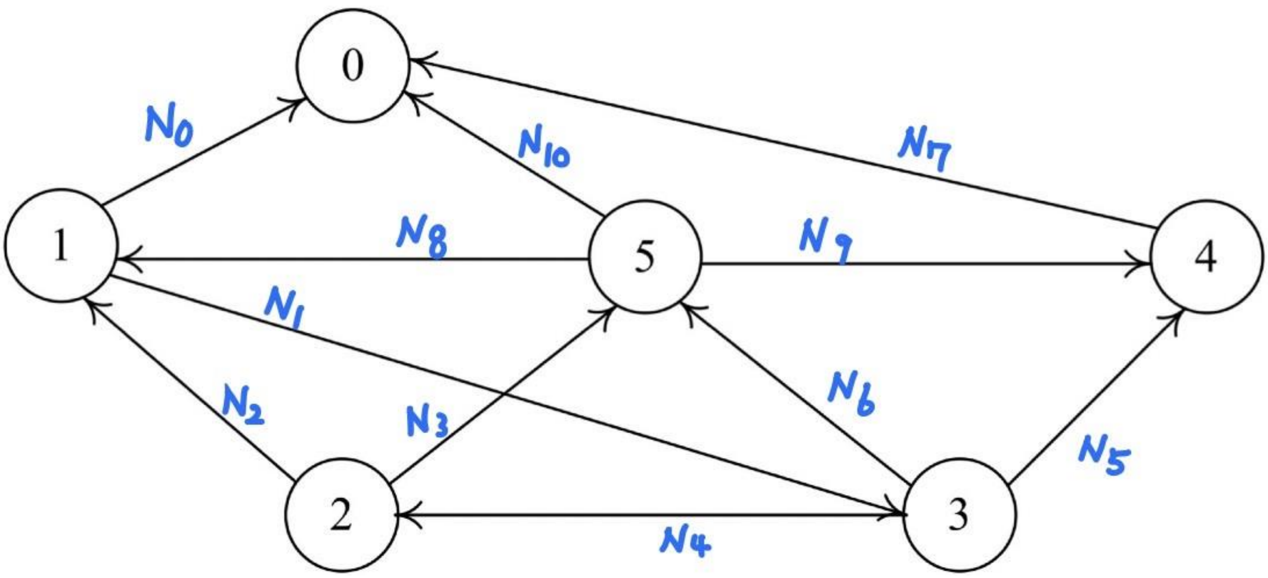
This column and row is indexed by (0~5) and (0~5), the graph is generated by LaTeX

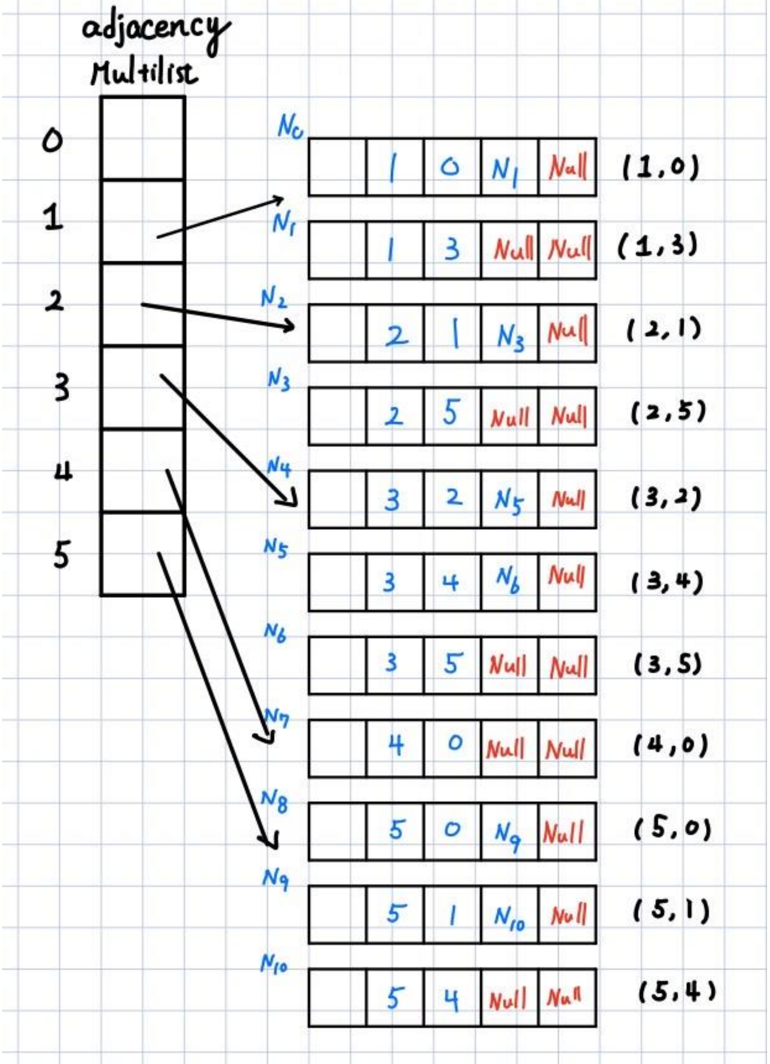
$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

(c)



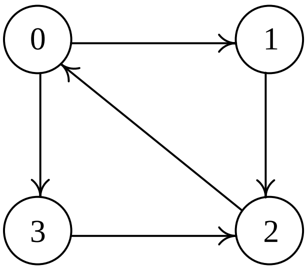
(d)





adjacency Multilist
0
1
2
3
4
5
N0 | 1 | 0 | N1 | Null | (1,0)
N1 | 1 | 3 | Null | Null | (1,3)
N2 | 2 | 1 | N3 | Null | (2,1)
N3 | 2 | 5 | Null | Null | (2,5)
N4 | 3 | 2 | N5 | Null | (3,2)
N5 | 3 | 4 | N6 | Null | (3,4)
N6 | 3 | 5 | Null | Null | (3,5)
N7 | 4 | 0 | Null | Null | (4,0)
N8 | 5 | 0 | N9 | Null | (5,0)
N9 | 5 | 1 | N10 | Null | (5,1)
N10 | 5 | 4 | Null | Null | (5,4)

3. (10%) Is the digraph below strongly connected? List all the simple paths.



<answer>

Yes, the graph is strongly connected.

Simple path of the graph:

Starting from 0

0 → 1
0 → 3
0 → 1 → 2
0 → 3 → 2
0 → 1 → 2 → 0
0 → 3 → 2 → 0

Starting from 1

1 → 2
1 → 2 → 0
1 → 2 → 0 → 1
1 → 2 → 0 → 3

Starting from 2

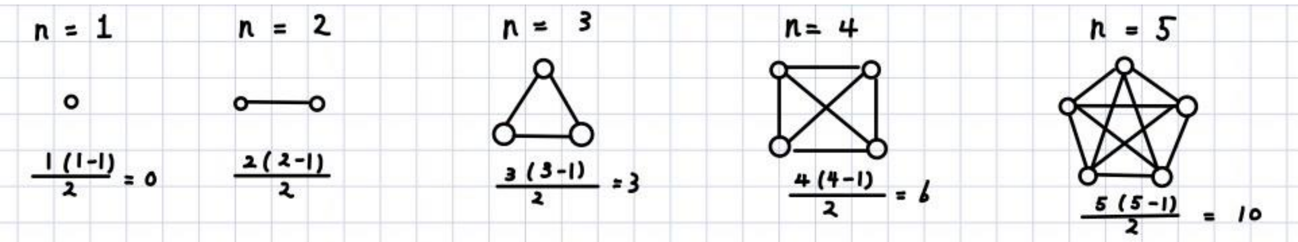
2 → 0
2 → 0 → 1
2 → 0 → 3
2 → 0 → 1 → 2
2 → 0 → 3 → 2

Starting from 3.

3 → 2
3 → 2 → 0
3 → 2 → 0 → 3
3 → 2 → 0 → 1

4. (10%) Draw the complete undirected graphs on one, two, three, four, and five vertices. Prove that the number of edges in an n-vertex complete graph is n(n-1)/2.

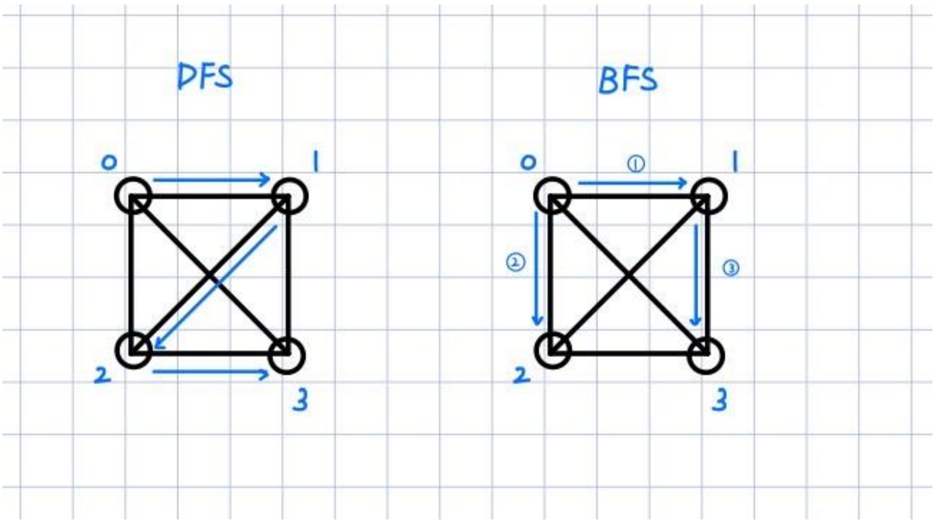
<answer>



For each n vertex to be connected to other vertex there are n – 1 edges, there are n vertices so there are $\frac{n \times (n-1)}{2}$ edges, since 2 of each are replicates.

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.

<answer>



Both the order of traversal for DFS and BFS are $0 \rightarrow 1 \rightarrow 2 \rightarrow 3$

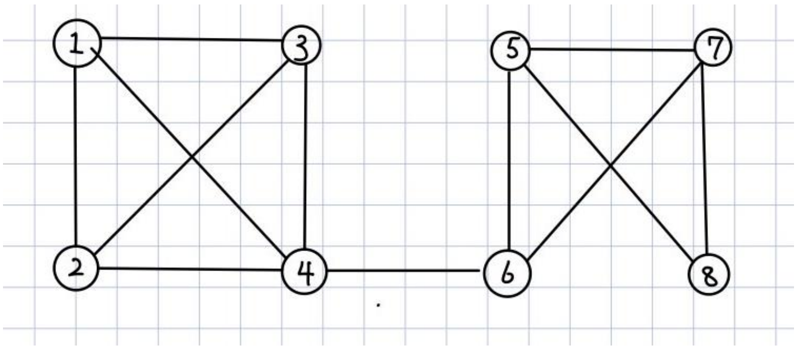
6. (6%) Let *G* be a graph whose vertices are the integers 1 through 8, and let the adjacent vertices of each vertex be given by the table below:

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)

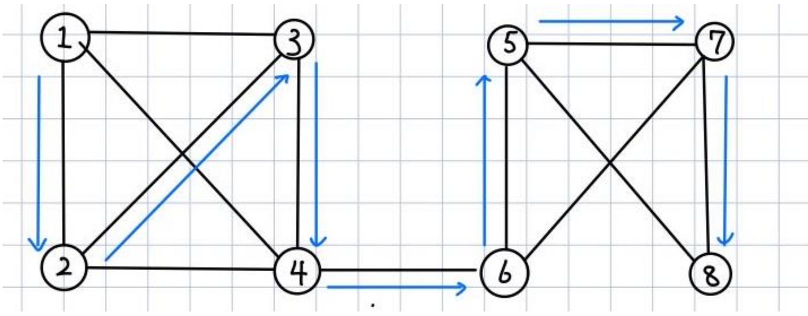
Assume that, in a traversal of *G*, the adjacent vertices of a given vertex are returned in the same order as they are listed in the table above.

<answer>

(a) Draw *G*.

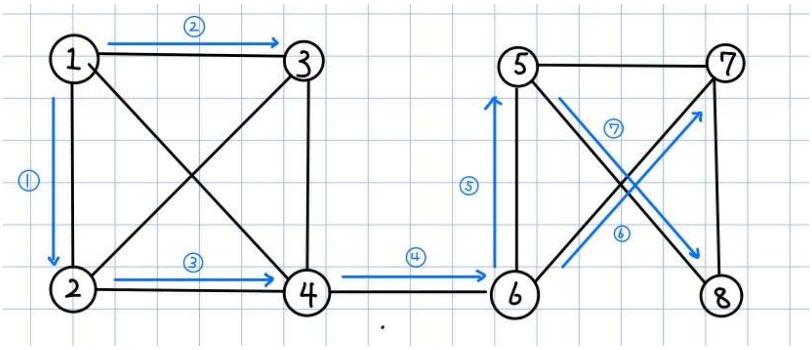


(b) Give the sequence of vertices of *G* visited using a DFS traversal starting at vertex 1.



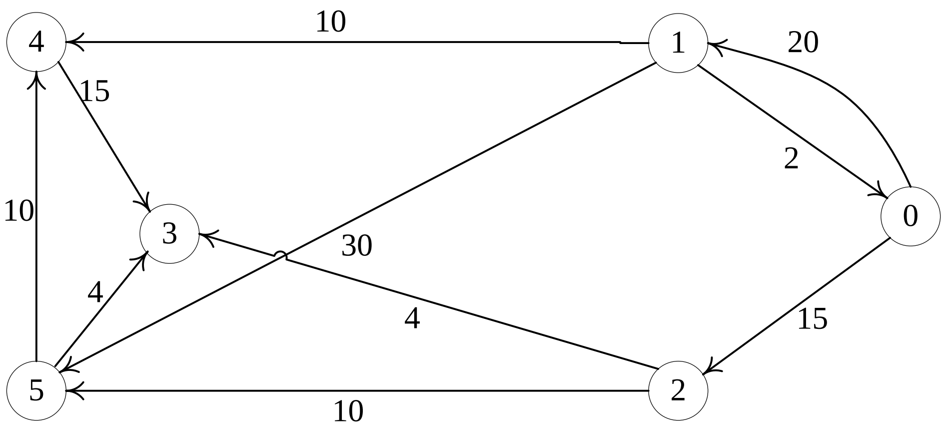
1 → 2 → 3 → 4 → 6 → 5 → 7 → 8

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



1 → 2 → 3 → 4 → 6 → 5 → 7 → 8

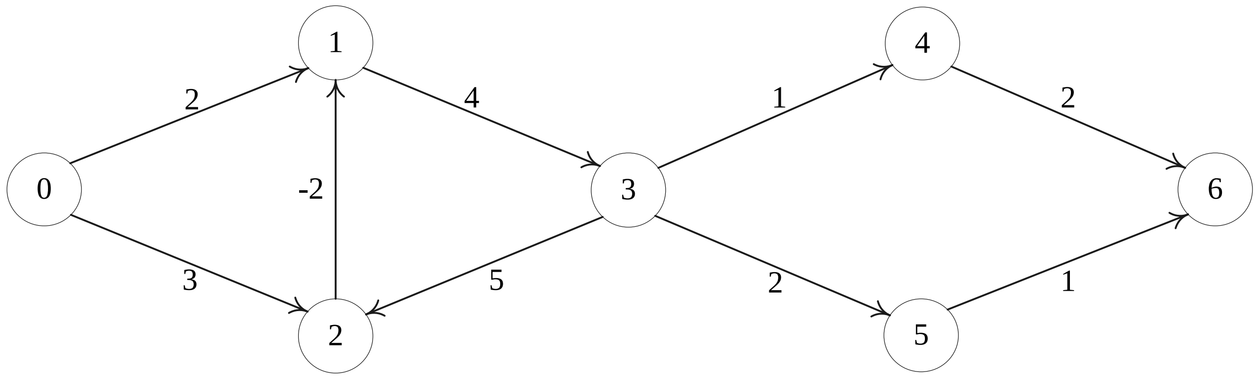
7. (10%) Use ShortestPath (Program 6.8) (Dijkstra's algorithm) to obtain, in nondecreasing order, the lengths and the paths of the shortest paths from vertex 0 to all remaining vertices in the graph below.



<answer>

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

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?

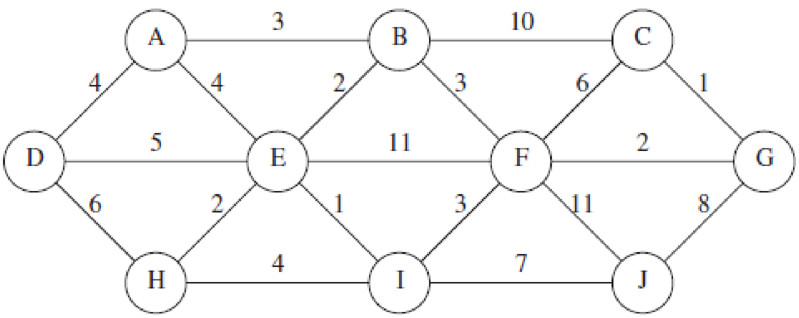


<answer>

The shortest path from 0 to 6 is 0 → 2 → 1 → 3 → 4 → 6 with length 8.

The Dijkstra's algorithm wouldn't work because, negative weight will not guarantee a closed vertex is indeed minimal throughout the traversal.

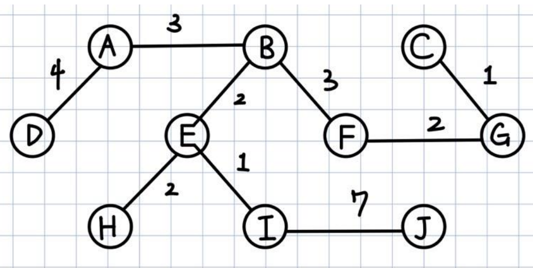
9. (10%) For the weighted graph G shown below,



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

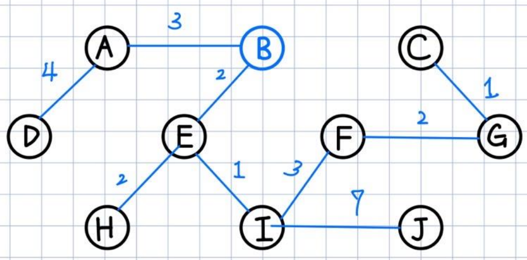
<answer>

Kruskal's



Prim's

Suppose we pick B as first vertex in tree



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

<answer>

No, the minimum spanning tree is not unique, it varies with tie-break mechanism if same weight edge occurs, but the total cost shouldn’t vary for distinct minimal spanning tree.

10. (10%) Does the following set of precedence relations (<) define a partial order on the elements 0 through 4? Why?

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

<answer>

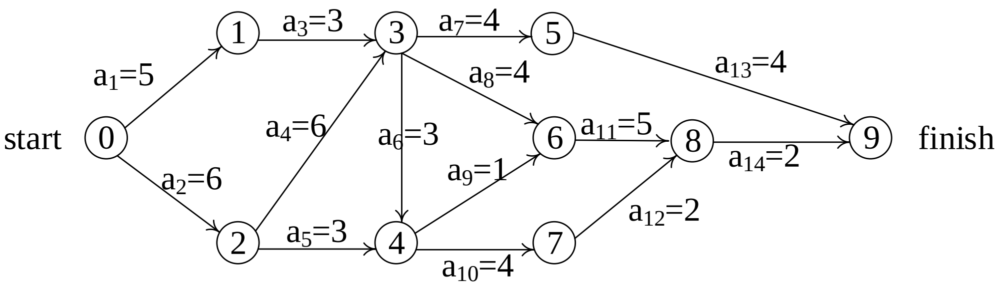
Assume a partial order is defined on the relations, since partial order guarantees transitive, thus from above relations, we get $0 < 0$ which is a mapping from 0 to 0, which violates the irreflexive properties of partial order. Thus, it doesn't define a partial order.

11. (10%) 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.

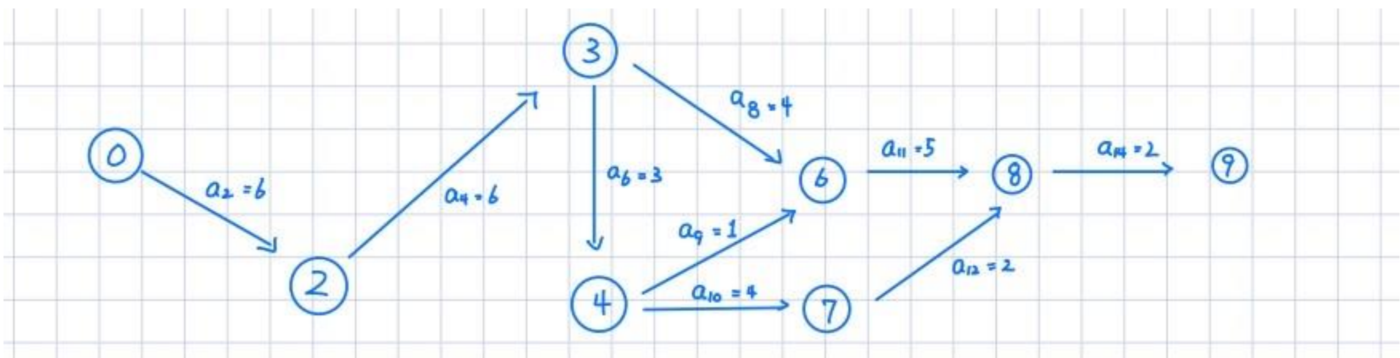
(b) What is the earliest time the project can finish?

(c) Which activities are critical? Fill the table below for answers to (a), (b), and (c).



activity	Early time	Late time	slack	critical
	$e(a_i)$	$l(a_i)$	$l(a_i) - e(a_i)$	
a_1	0	4	4	No
a_2	0	0	0	Yes
a_3	5	9	4	No
a_4	6	6	0	Yes
a_5	6	12	6	No
a_6	12	12	0	Yes
a_7	12	15	3	No
a_8	12	12	0	Yes
a_9	15	15	0	Yes
a_{10}	15	15	0	Yes
a_{11}	16	16	0	Yes
a_{12}	19	19	0	Yes
a_{13}	16	19	3	No
a_{14}	21	21	0	Yes

(d) Is there any single activity whose speed-up would result in a reduction of the project finish time?



The single activity that reduce project finish time is a_2, a_4, a_{14}, since they are on all critical path.