**EECS2040 Data Structure Hw #5 (Chapter 6 Graph)**

**due date 5/30/2022**

**by 108011235 陳昭維**

**(Part 1: 2% of final Grade)**

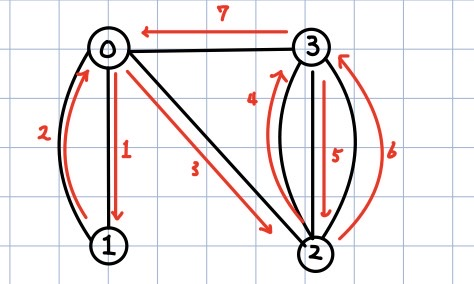
**Part 1**

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



<answer>

Yes there is an Eulerian walk



1. (10%) For the digraph below obtain
2. The in-degree and out-degree of each vertex
3. Its adjacency-matrix
4. Its adjacency-list representation
5. Its strongly connected components



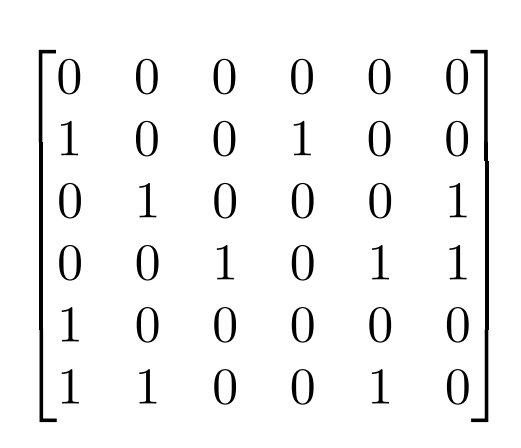
<answer>

(a)

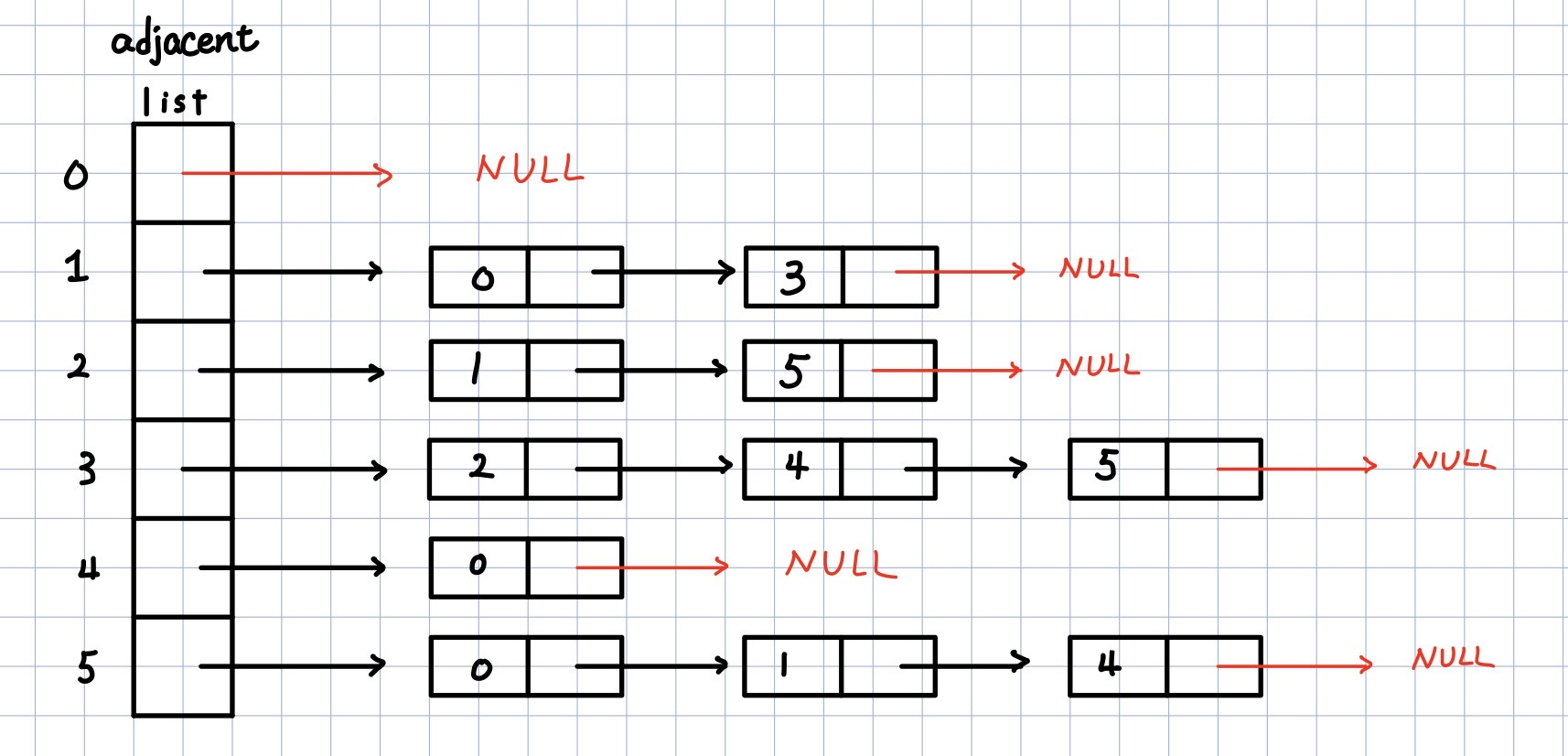
|  |  |  |
| --- | --- | --- |
| Vertex | In-degree | Out-degree |
| 0 | 3 | 0 |
| 1 | 2 | 2 |
| 2 | 1 | 2 |
| 3 | 1 | 3 |
| 4 | 2 | 1 |
| 5 | 2 | 3 |

(b)

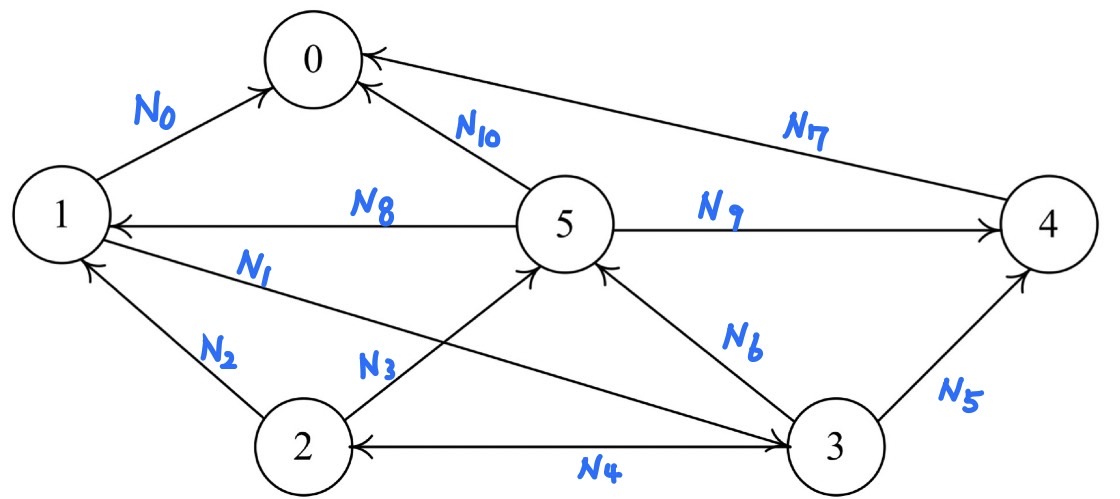
This column and row is indexed by (0~5) and (0~5), the graph is generated by LaTeX

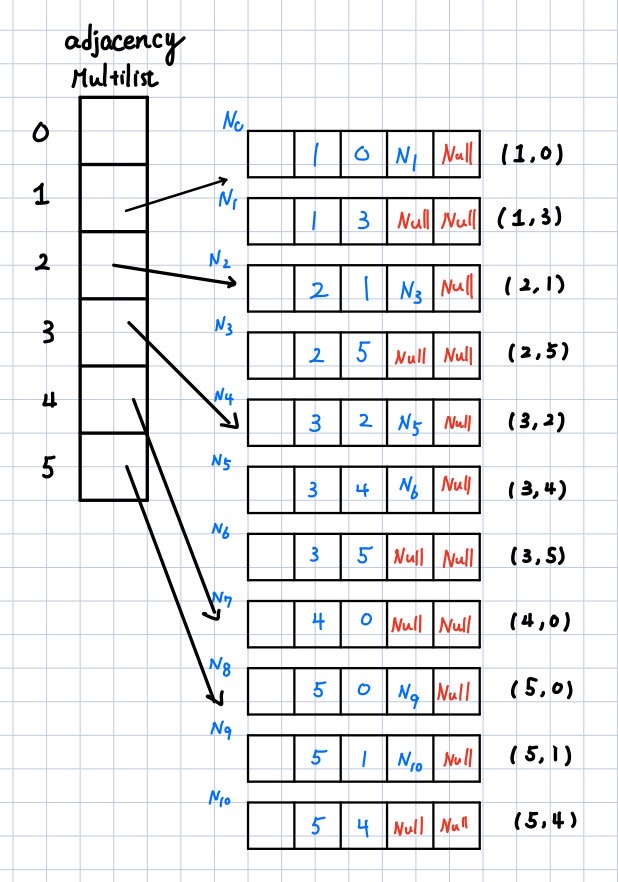


(c)



(d)





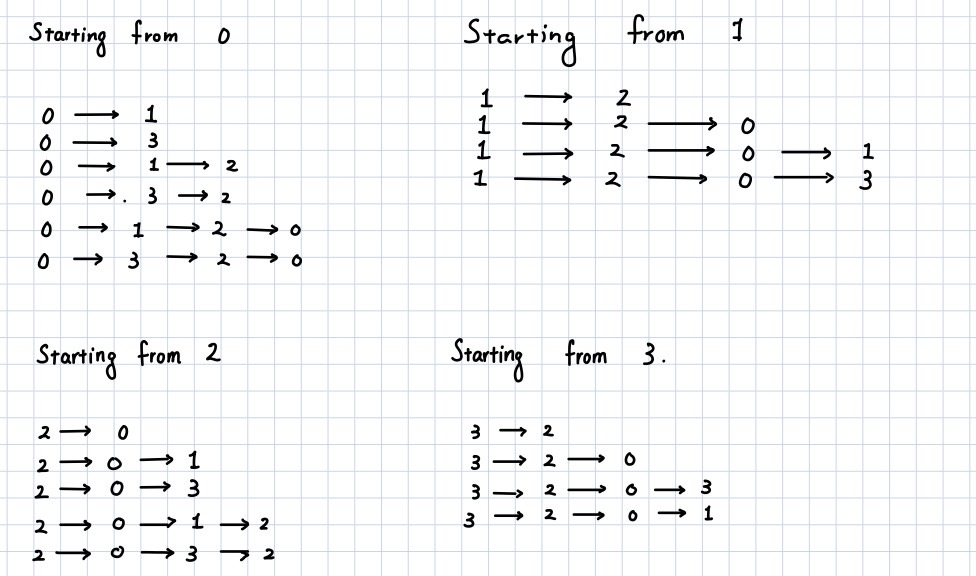
1. (10%) Is the digraph below strongly connected? List all the simple paths.



<answer>

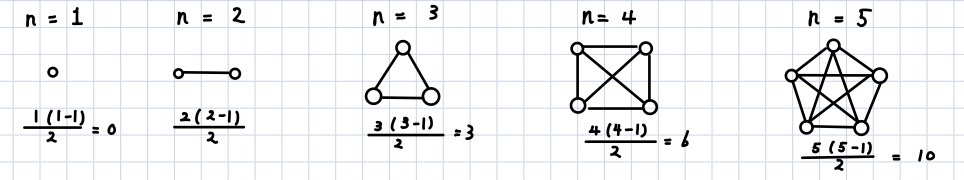
Yes, the graph is strongly connected.

Simple path of the graph:



1. (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.

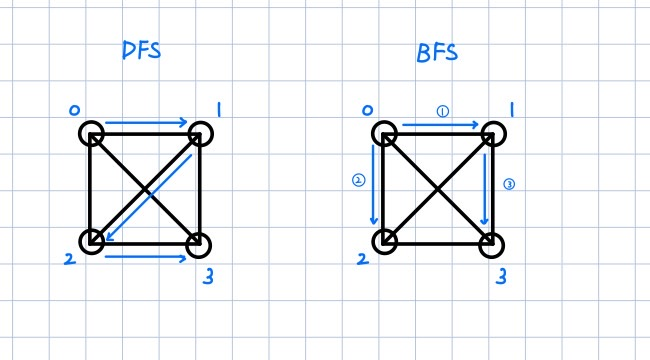
<answer>



For each n vertex to be connected to other vertex there are n – 1 edges, there are n vertices so there are edges, since 2 of each are replicates.

1. (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

1. (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)

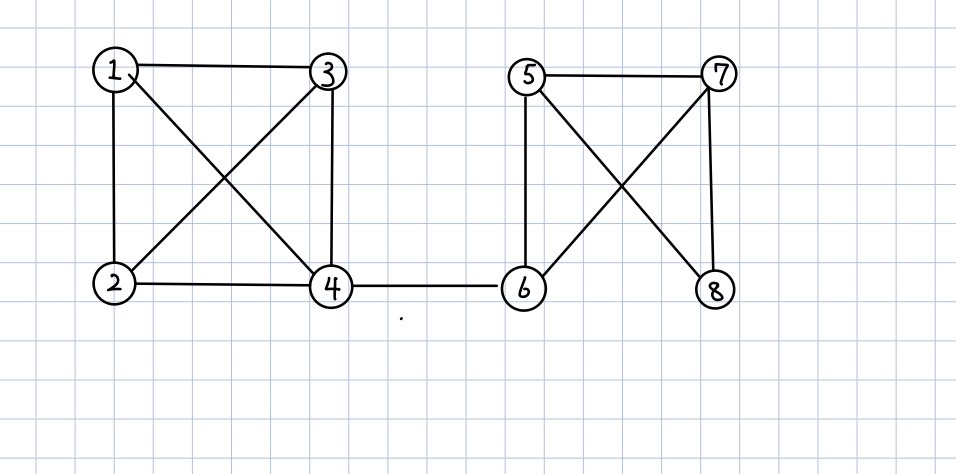
7 (5, 6, 8)

8 (5, 7)

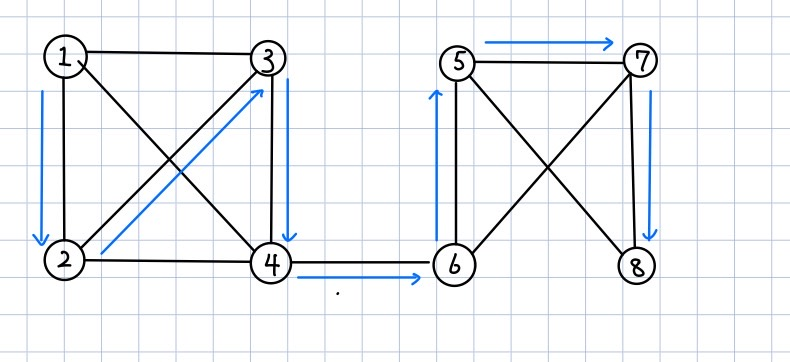
Assume that, in a traversal of *G*, the adjacent vertices of a given vertex are returned in the same order as they are listed in the table above.

<answer>

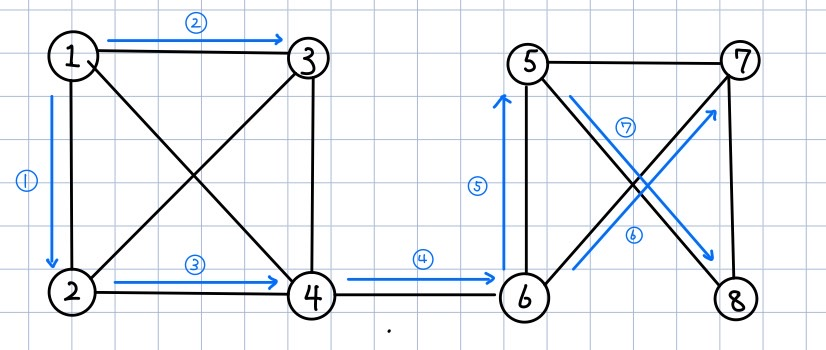
(a) Draw *G*.



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



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



1. (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 |  | 20 |
| 2 |  | 15 |
| 3 |  | 19 |
| 4 |  | 30 |
| 5 |  | 25 |

1. (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 with length 8.

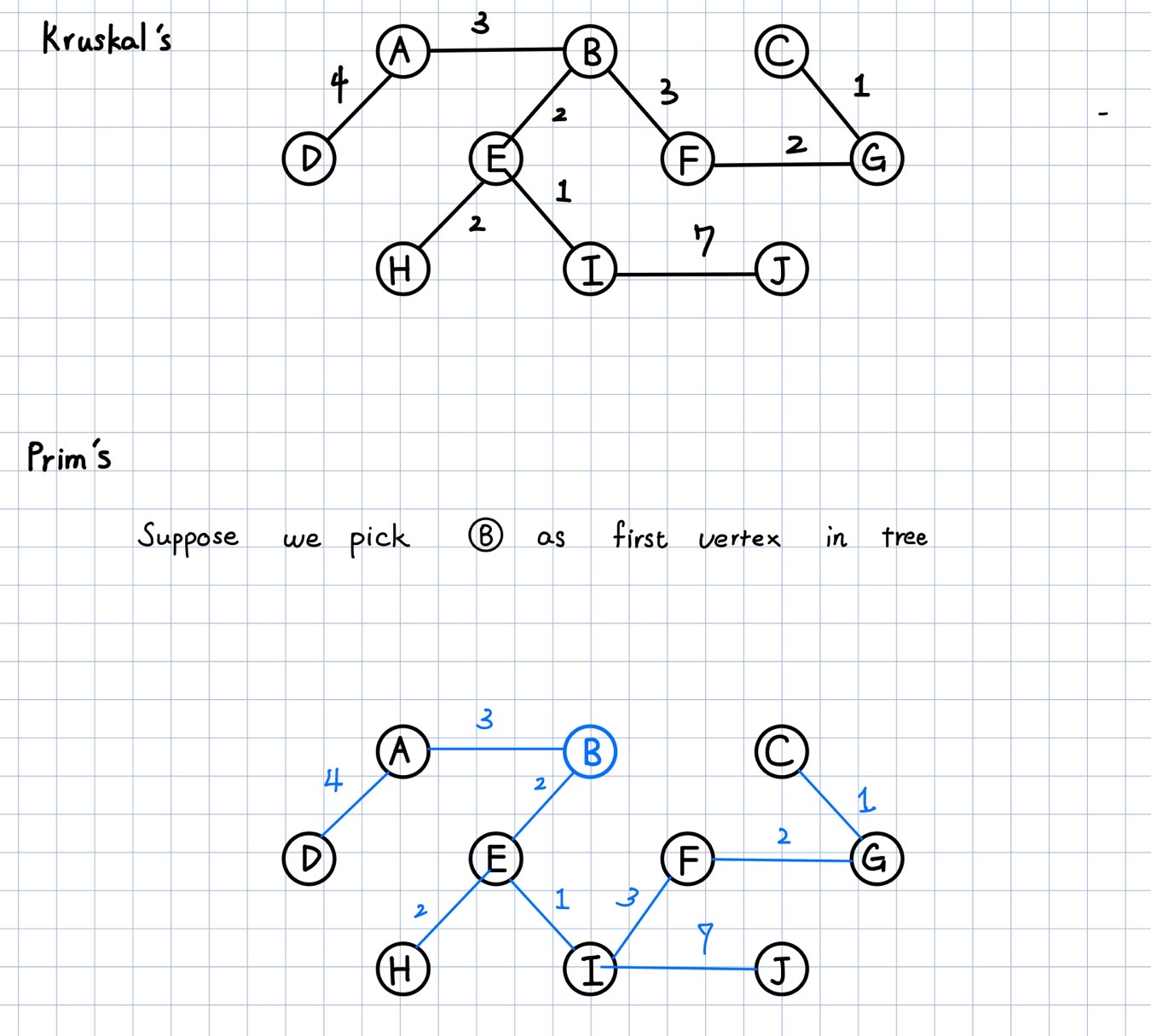
The Dijkstra’s algorithm wouldn’t work because, negative weight will not guarantee a closed vertex is indeed minimal throughout the traversal.

1. (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>



(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.

1. (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>

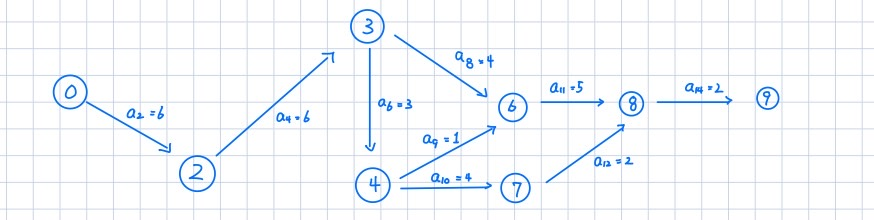
Assume a partial order is defined on the relations, since partial order guarantees transitive, thus from above relations, we get 0 < 0 which is a mapping from 0 to 0, which violates the irreflexive properties of partial order. Thus, it doesn’t define a partial order.

1. (10%) For the AOE network shown below,
2. Obtain the early, e(ai), and late, l(ai), start times for each activity. Use the forward-backward approach.
3. What is the earliest time the project can finish?
4. Which activities are critical? Fill the table below for answers to (a), (b), and (c).



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| activity | Early time | Late time | slack | critical |
| e(ai) | l(ai) | l(ai) - e(ai) |  |
| a1 | 0 | 4 | 4 | No |
| a2 | 0 | 0 | 0 | Yes |
| a3 | 5 | 9 | 4 | No |
| a4 | 6 | 6 | 0 | Yes |
| a5 | 6 | 12 | 6 | No |
| a6 | 12 | 12 | 0 | Yes |
| a7 | 12 | 15 | 3 | No |
| a8 | 12 | 12 | 0 | Yes |
| a9 | 15 | 15 | 0 | Yes |
| a10 | 15 | 15 | 0 | Yes |
| a11 | 16 | 16 | 0 | Yes |
| a12 | 19 | 19 | 0 | Yes |
| a13 | 16 | 19 | 3 | No |
| a14 | 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 a2 , a4, a14, since they are on all critical path.