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

**due date 5/30/2022 (Part 1: 2% of final Grade) by 109070025 林泓錩**

**Part 1**

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



**Ans:**

Yes, 0 =>1 =>0 =>2 =>3 =>2 =>3 =>0.

1. (10%) For the digraph below obtain
2. The in-degree and out-degree of each vertex

**Ans:**

Vertex0: in-degree: 3; out-degree: 0

Vertex1: in-degree: 2; out-degree: 2

Vertex2: in-degree: 1; out-degree: 2

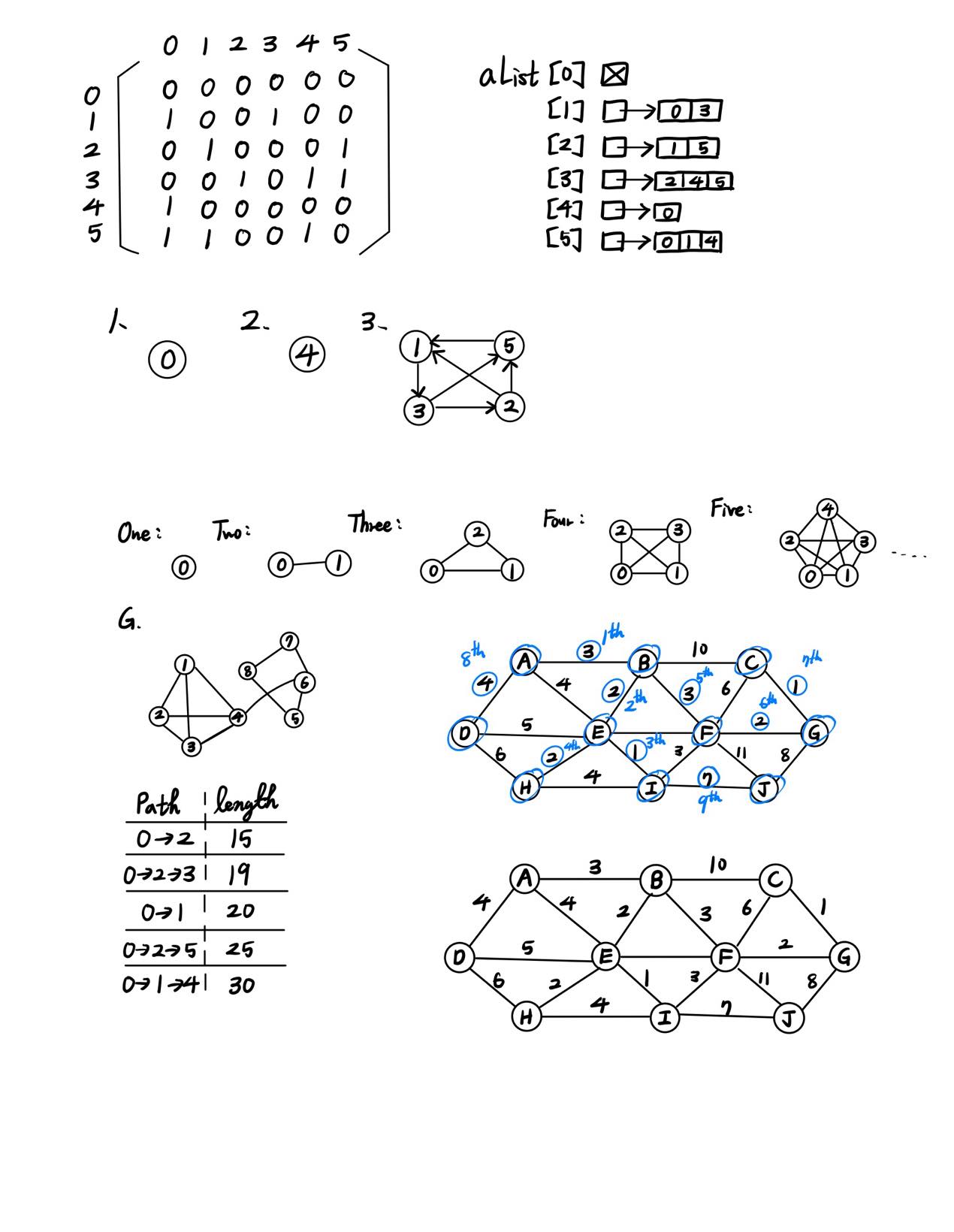
Vertex3: in-degree: 1; out-degree: 3

Vertex4: in-degree: 2; out-degree: 1

Vertex5: in-degree: 2; out-degree: 3

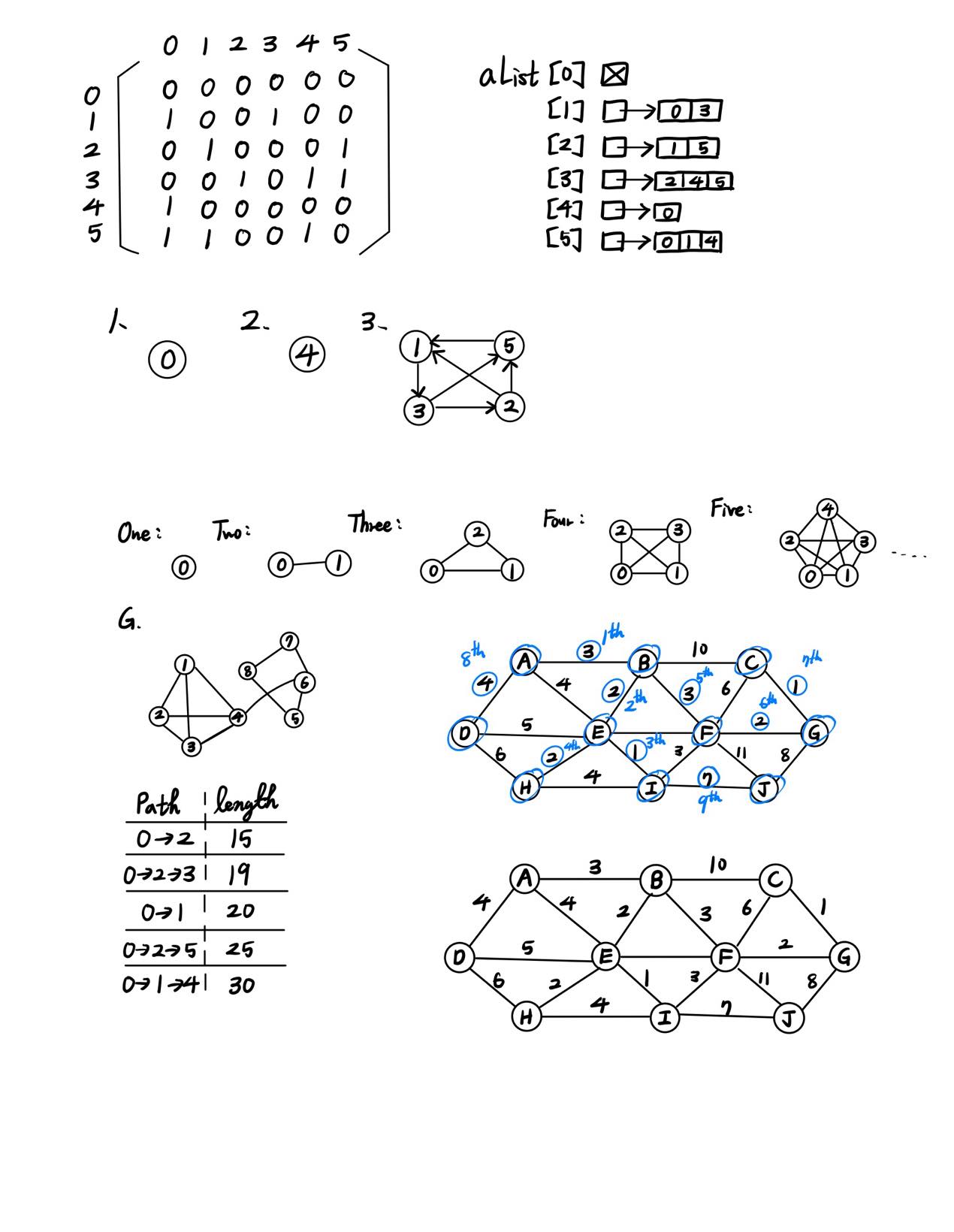
1. Its adjacency-matrix

**Ans:**

****

1. Its adjacency-list representation

**Ans:**

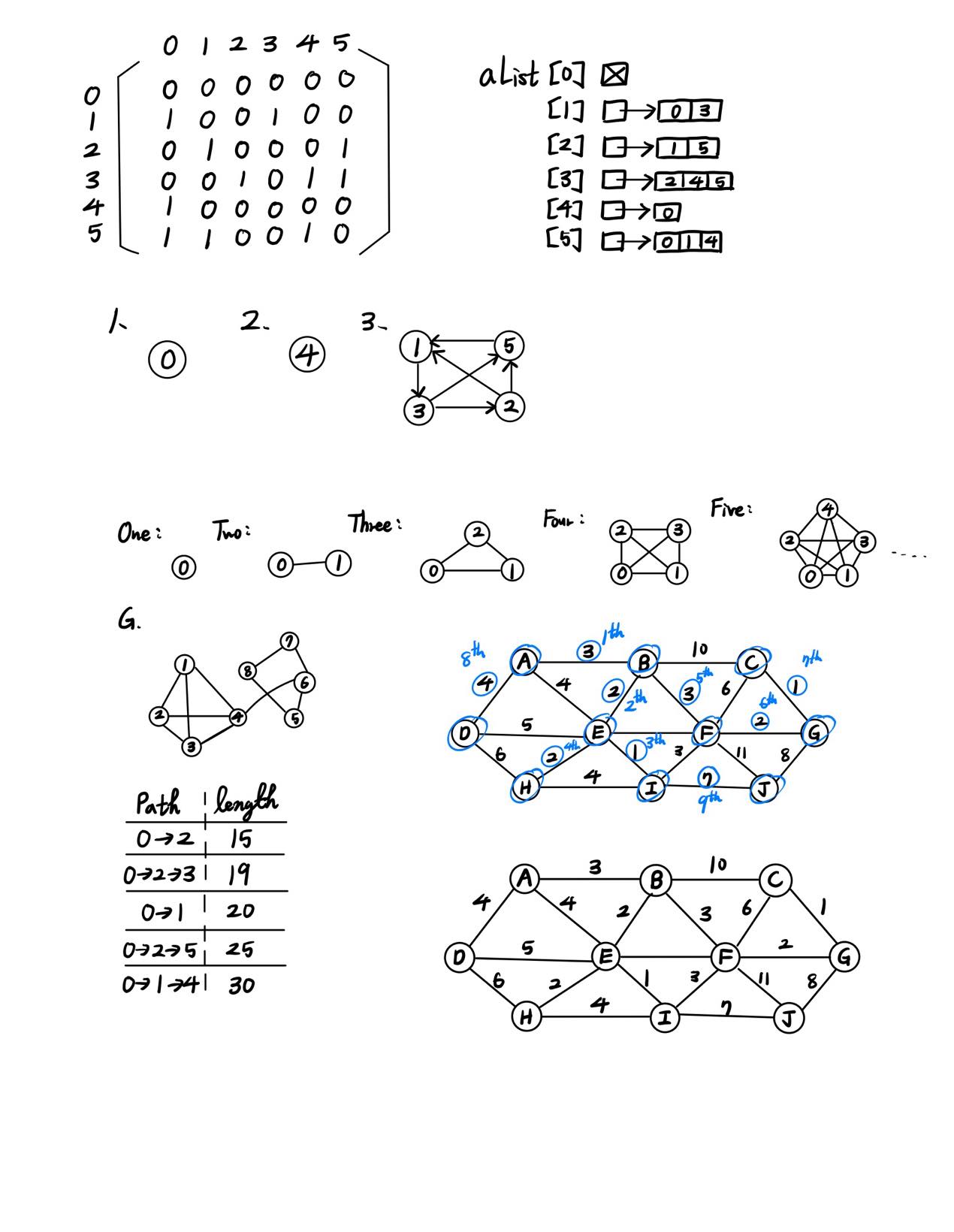


1. Its strongly connected components



**Ans:**

Three strongly connected components.



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



**Ans:**

Yes, it is strongly connected.

From 0 to 1: 0 =>1 ; From 0 to 2: 0 =>1 =>2 and 0 =>3 =>2 ; From 0 to 3: 0 =>3

From 1 to 0: 1 =>2 =>0 ; From 1 to 2: 1 =>2 ; From 1 to 3: 1 =>2 =>0 =>3

From 2 to 0: 2 =>0 ; From 2 to 1: 2 =>0 =>1 ; From 2 to 3: 2 =>0 =>3

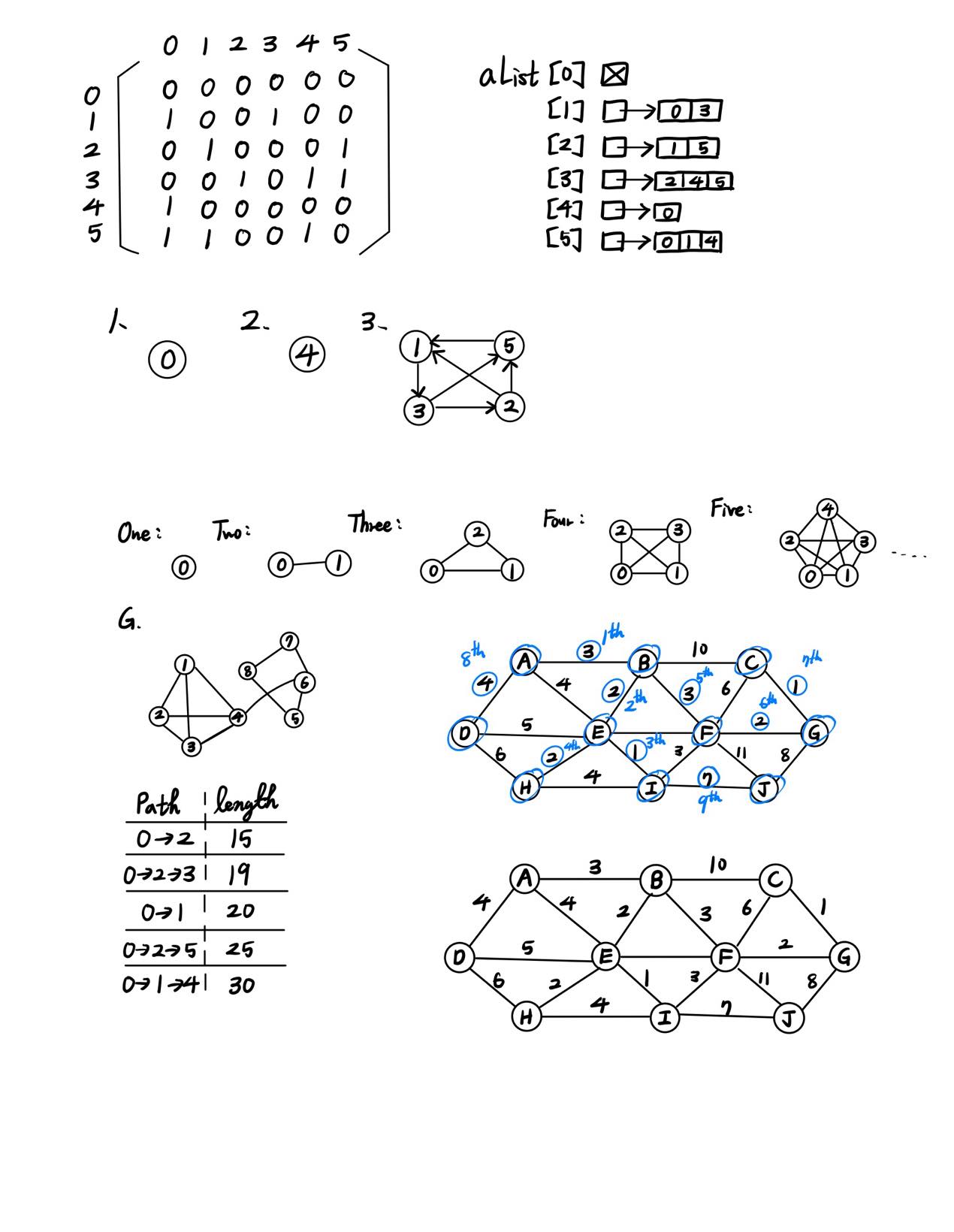
From 3 to 0: 3 =>2 =>0 ; From 3 to 1: 3 =>2 =>0 =>1 ; From 3 to 2: 3 =>2

From 0 to 0: 0 =>1 =>2 =>0 and 0 =>3 =>2 =>0 From 1 to 1: 1 =>2 =>0 =>1

From 2 to 2: 2 =>0 =>1 =>2 and 2 =>0 =>3 =>2 From 3 to 3: 3 =>2 =>0 =>3

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.

**Ans:**

****

Assume there are n vertex, each vertex can connect n-1vertex. Each edge connect two vertex. Therefore, the amount of edge is n(n−1)/2.

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.

**Ans:**

Depth-first search: 0 =>1 =>2 =>3

Breadth-first search: 0 =>1 =>2 =>3

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)

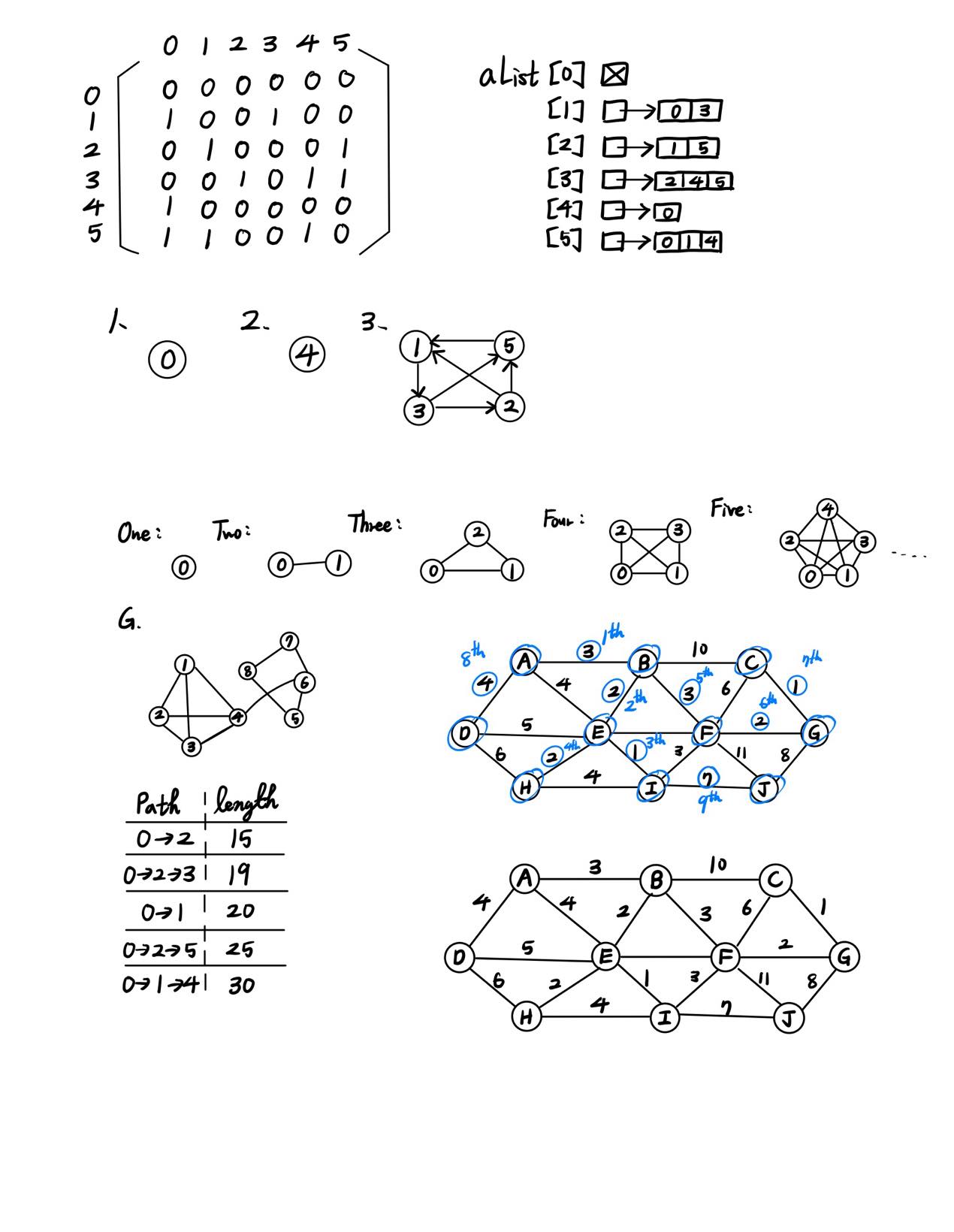
7 (5, 6, 8)

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.

1. Draw *G*.

**Ans:**



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

**Ans:**

1 =>2 =>3 =>4 =>6 =>5 =>7 =>8

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

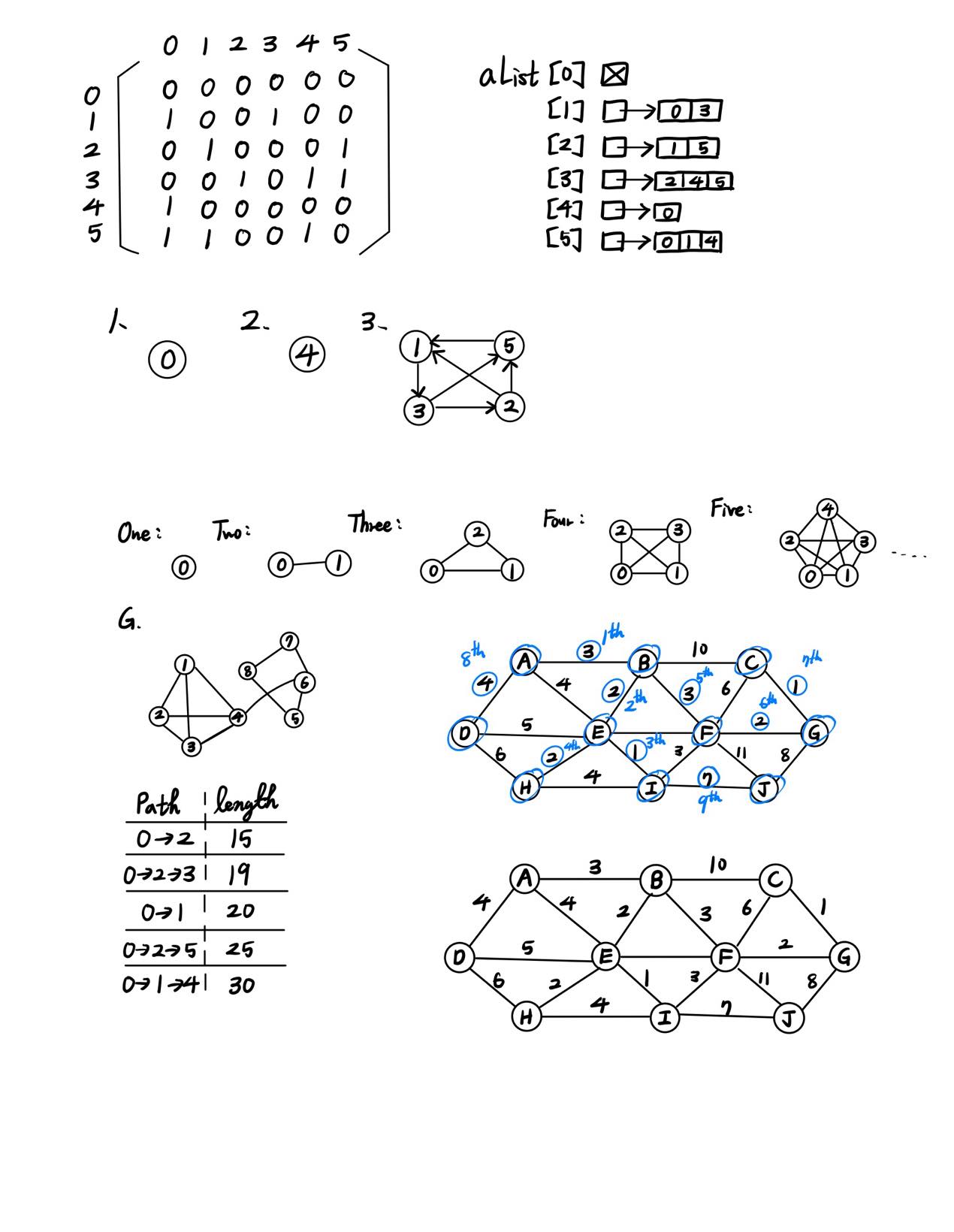
**Ans:**

1 =>2 =>3 =>4 =>6 =>5 =>7 =>8

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.



**Ans:**

****

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?



**Ans:**

The shortest path is 0 =>2 =>1 =>3 =>4 =>6 or 0 =>2 =>1 =>3 =>5 =>6, the length is 8, for

The grapf has the negative path length, hence, the ShortPath couldn’t operate normally, there will be an error.

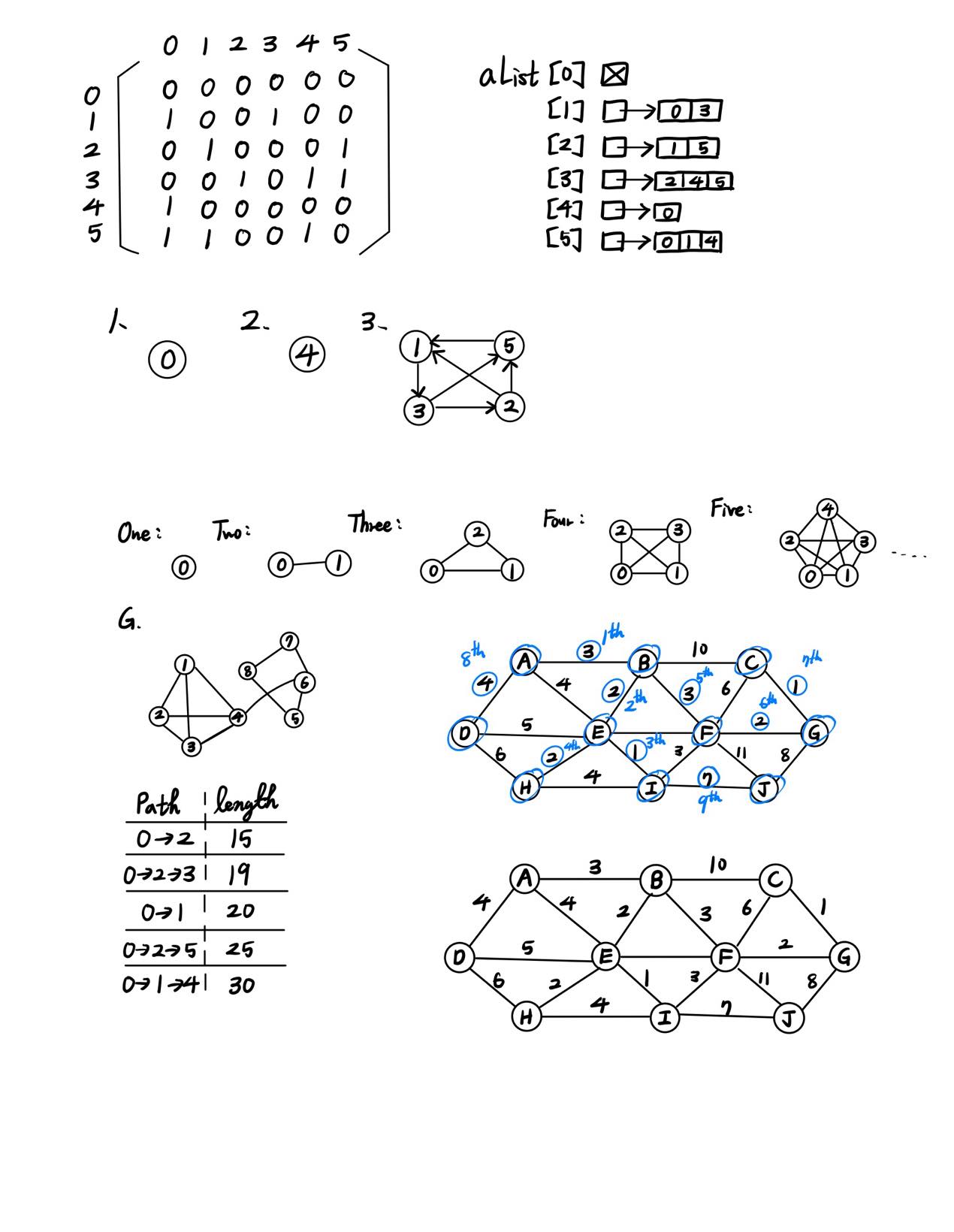
1. (10%) For the weighted graph G shown below,



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

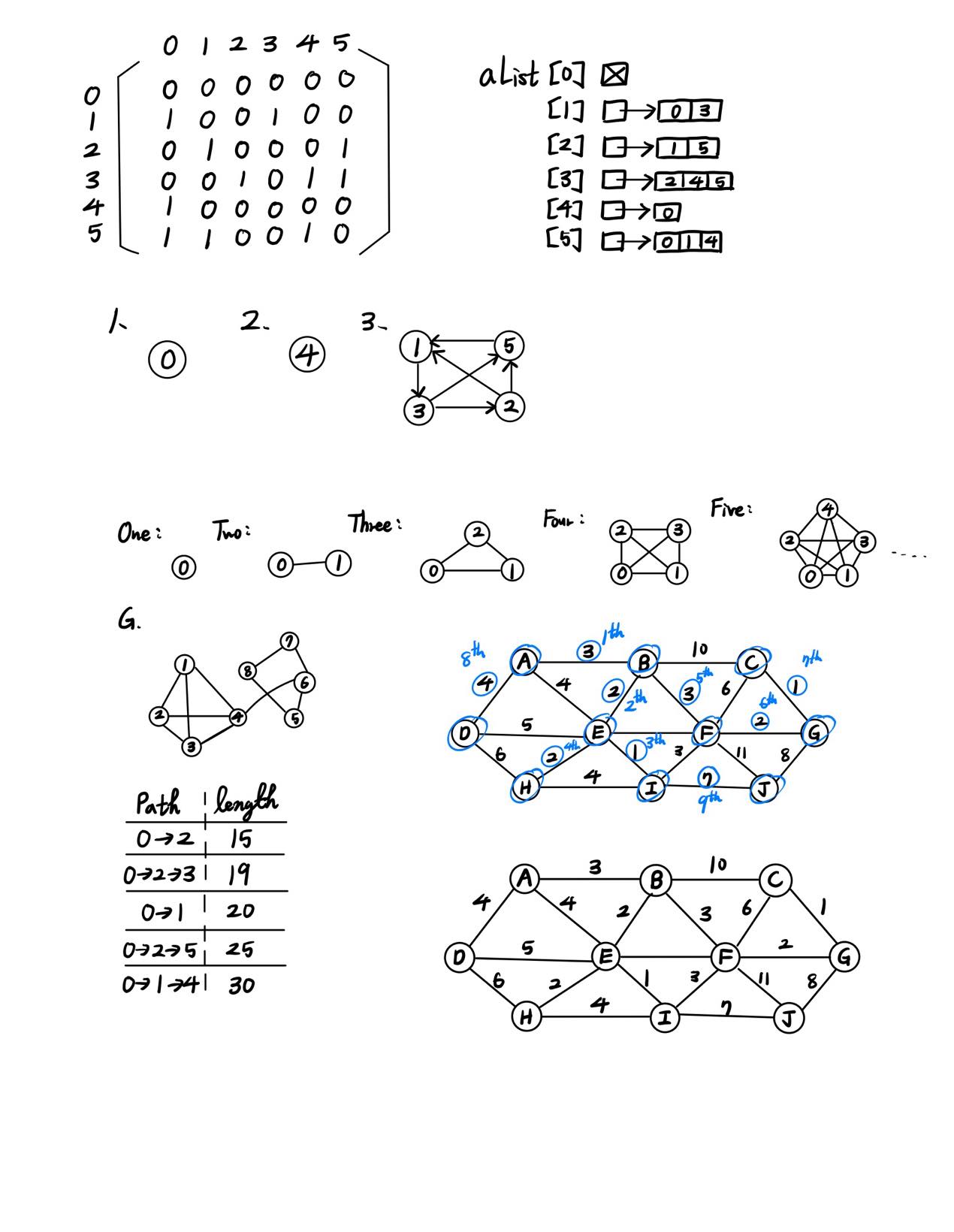
**Ans:**

Start from A.



1. Is this minimum spanning tree unique? Why?

**Ans:**



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

**Ans:**

No, we know 0<1, 1<2, 2<4, by transitive relation, we know 0<4, but question say that 4<0, it’s contradiction.

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



**Ans:**

speed up a2.a4.a14, since they are on single path.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| activity | Early time | Late time | slack | critical |
| e(ai) | l(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 |