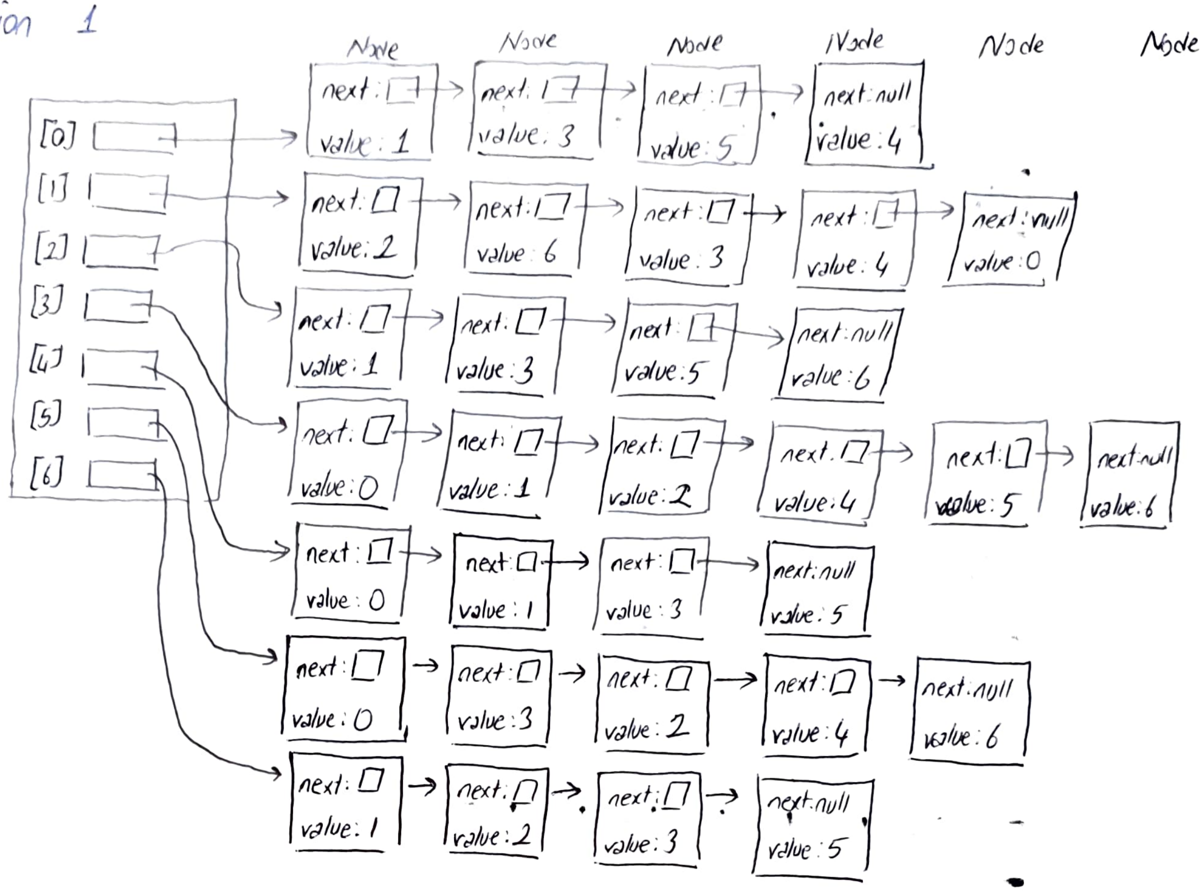
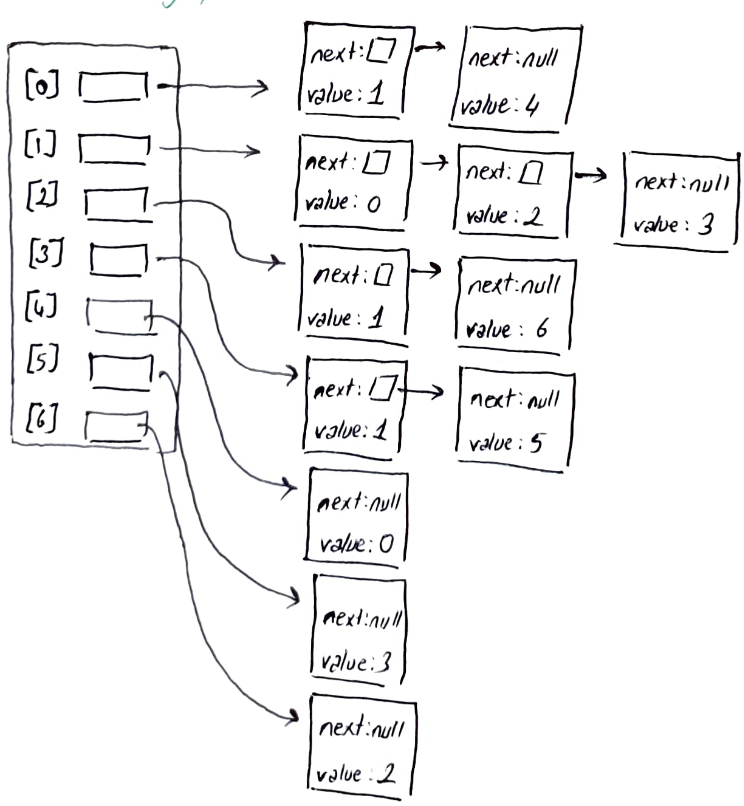
Emirhan Uzun
171044019

Question 1

1.a)



2nd graph



Emirhan Uzun
171044019

1.b) 1^{st} Graph

Columns

Rows

	[0]	[1]	[2]	[3]	[4]	[5]	[6]
[0]		1.0		1.0	1.0	1.0	
[1]	1.0		1.0	1.0	1.0		1.0
[2]		1.0		1.0		1.0	1.0
[3]	1.0	1.0	1.0		1.0	1.0	1.0
[4]	1.0	1.0		1.0		1.0	
[5]	1.0		1.0	1.0	1.0		1.0
[6]		1.0	1.0	1.0		1.0	

2^{nd} Graph

	[0]	[1]	[2]	[3]	[4]	[5]	[6]
[0]		1.0			1.0		
[1]	1.0		1.0	1.0			
[2]		1.0					1.0
[3]		1.0				1.0	
[4]	1.0						
[5]				1.0			
[6]			1.0				

Emirhan Uzun
171044019

1.c) For first graph
$|V| = n = 7$
$|E| = m = 16$

Density is ratio of m to n^2
$D = \frac{16}{49}$

For second graph
$|V| = n = 7$
$|E| = m = 6$

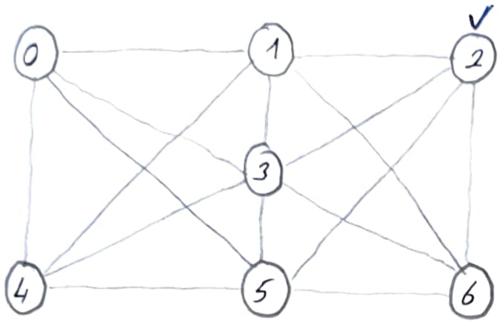
$D = \frac{m}{n^2} = \frac{6}{49}$

For first graph, number of edge is bigger than vertices. That means this graph is dense graph. Adjacency matrix is appropriate for that graph. Because adjacency matrix is good for dense graph.
For second graph, edge is less than vertices. This graph is sparse graph. Adjacency list is appropriate for sparse graphs. Because it gives better performance.

For first graph, because of matrix is used, there is no wasted space so much. For second graph, storage occurs because adjacency matrix is 0,125 fill.

Emirhan Uzun
171044019

1. d) 1st Graph

1- 

Visit Order: 2
Finish Order:

2- Traverse largest to smallest. So now, we go to the vertex 6.

Visit order: 2,6
Finish order:

3- From 6 to 5 (Largest to smallest)

Visit order: 2,6,5
Finish order:

4- Now in (0,3,4), we go to the 4

Visit order: 2,6,5,4
Finish order:

5- We are at vertex 4. Our not being visited vertex are 0, 1, 3.
So we go to the vertex 3

Visit order: 2,6,5,4, 3
Finish order:

6- From vertex 3, we go to the 1. Because 1 is greater than 0.

Visit order: 2,6,5,4,3,1
Finish order:

Emirhan Uzun
171044019

7- Now we go to the 0.
Visit order: 2, 6, 5, 4, 3, 1, 0
Finish order:

8- There is no being visited vertex. So we mark 0 as visited and return 1
Visit order: 2, 6, 5, 4, 3, 1, 0
Finish order: 0

9- Mark 1 visited and return 3 (Because all adjacent nodes to 1 being visited)
Visit order: 2, 6, 5, 4, 3, 1, 0
Finish order: 0, 1

10- Mark 3 visited and return 4 (All nodes to 3 being visited)
Visit order: 2, 6, 5, 4, 3, 1, 0
Finish order: 0, 1, 3

11- Mark 4 visited and return 5 (All nodes to 4 being visited)
Visit order: 2, 6, 5, 4, 3, 1, 0
Finish order: 0, 1, 3, 4

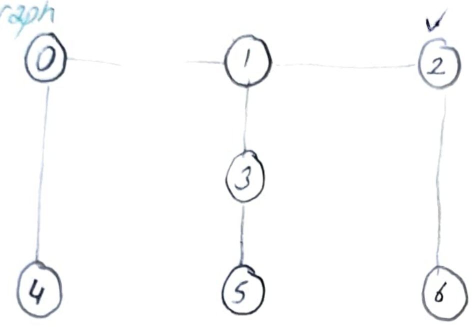
12- Mark 5 visited and return 6 (All nodes to 5 being visited)
Visit order: 2, 6, 5, 4, 3, 1, 0
Finish order: 0, 1, 3, 4, 5

13- Mark 6 and return 2 (All nodes to 6 being visited)
Visit order: 2, 6, 5, 4, 3, 1, 0
Finish order: 0, 1, 3, 4, 5, 6

14- There are no adjacent to not being visited. Mark 2 and finish.
Visit order: 2, 6, 5, 4, 3, 1, 0
Finish order: 0, 1, 3, 4, 5, 6, 2

Emirhan Uzun
171044019

2nd Graph

1- 

Visit order : 2
Finish order :

2- Traverse largest to smallest. So now, we go to the vertex 6
Visit order : 2, 6
Finish order :

3- There are no vertices adjacent to 6. So we mark 6 as visited and return 2
Visit order : 2, 6
Finish order : 6

4- 1 is adjacent to 2 and not being visited
Visit order : 2, 6, 1
Finish order : 6

5- So now we can go 0 and 3. Traverse largest to smallest, so we go to the vertex 3
Visit order : 2, 6, 1, 3
Finish order : 6

6- We go the vertex 5.
Visit order : 2, 6, 1, 3, 5
Finish order : 6

7- There are no vertices adjacent to 5. So we mark 5 as visited and return 3.
Finish order : 6, 5

Emirhan Uzun
171044019

8- There is no more adjacent to not being visited. Mark 3 and return 1

Visit order: 2, 6, 1, 3, 5

Finish order: 6, 5, 3

9- 0 is not being visited.

Visit order: 2, 6, 1, 3, 5, 0

Finish order: 6, 5, 3

10- 4 is not being visited

Visit order: 2, 6, 1, 3, 5, 0, 4

Finish order: 6, 5, 3

11- There is no more adjacent to not being visited. Mark 4 and return 0

Visit order: 2, 6, 1, 3, 5, 0, 4

Finish order: 6, 5, 3, 4

12- There is no more adjacent to not being visited. Mark 0 and return 1

Visit order: 2, 6, 1, 3, 5, 0, 4

Finish order: 6, 5, 3, 4, 0

13- There is no more adjacent to not being visited. Mark 1 and return 2

Visit order: 2, 6, 1, 3, 5, 0, 4

Finish order: 6, 5, 3, 4, 0, 1

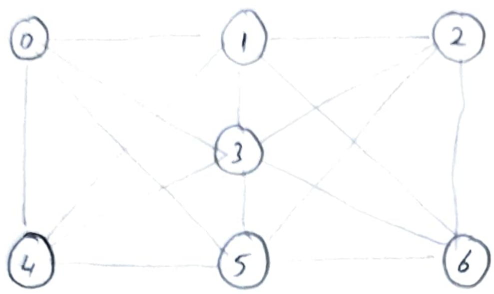
14- There is no more adjacent to not being visited. Mark 2 and finish

Visit order: 2, 6, 1, 3, 5, 0, 4

Finish order: 6, 5, 3, 4, 0, 1, 2

Emirhan Uzun
171044019

1.e) 1st Graph

1- 

Queue: 6, 5, 3, 1
Visit: 2

2- Now 2 is done and visit the node 6

Queue: 5,3,1
Visit: 2,6

3- 6's all adjacent has been visited and now 5 is visited.
Queue: 3,1,4,0
Visit: 2,6,5,

4- 5's adjacents are 4 and 0. We add to queue and now we visit the node 3.
Queue: 1,4,0
Visit: 2,6,5,3

5- 3's adjacents have been visited. So we are now in node 1
Queue: 4,0
Visit: 2,6,5,3,1

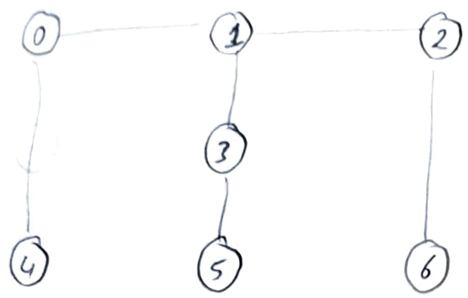
6- 1's adjacents have been visited. We are in node 4
Queue: 0
Visit: 2,6,5,3,1,4

7- We are in node 0 and it is last node. It has no adjacents
Queue: –
Visit: 2,6,5,3,1,4,0

Emirhan Uzun
171044019

2nd Graph

1-



Queue : 6, 1
Visit : 2

2- Now we done with 2 and go to the node 6.
Queue : 1
Visit : 2, 6

3- 6 has no adjacent. We are in node 1
Queue : 3, 0
Visit : 2, 6, 1

4- Node 1 has adjacent which is 3 and 0. Now in node 3;
Queue : 0, 5
Visit : 2, 6, 1, 3

5- Node 3 has adjacent which is 5. Now we are in node 0.
Queue : 5, 4
Visit : 2, 6, 1, 3, 0

6- Node 0 has adjacent with node 4. We add queue and now in node 5;
Queue : 4
Visit : 2, 6, 1, 3, 0, 5

7- Node 5 has no adjacent and now we are in last node which is 4
Queue : -
Visit : 2, 6, 1, 3, 0, 5, 4
4 has no adjacent and search is done.