

GTU Department of Computer Engineering
CSE 222/505 - Spring 2020
#Question 1

Mert Can BEŞİRLİ
1801042663

For each graph above, what are the $|V|=n$, the $|E|=m$, and the density? Which representation is better for each graph? Explain your answers.

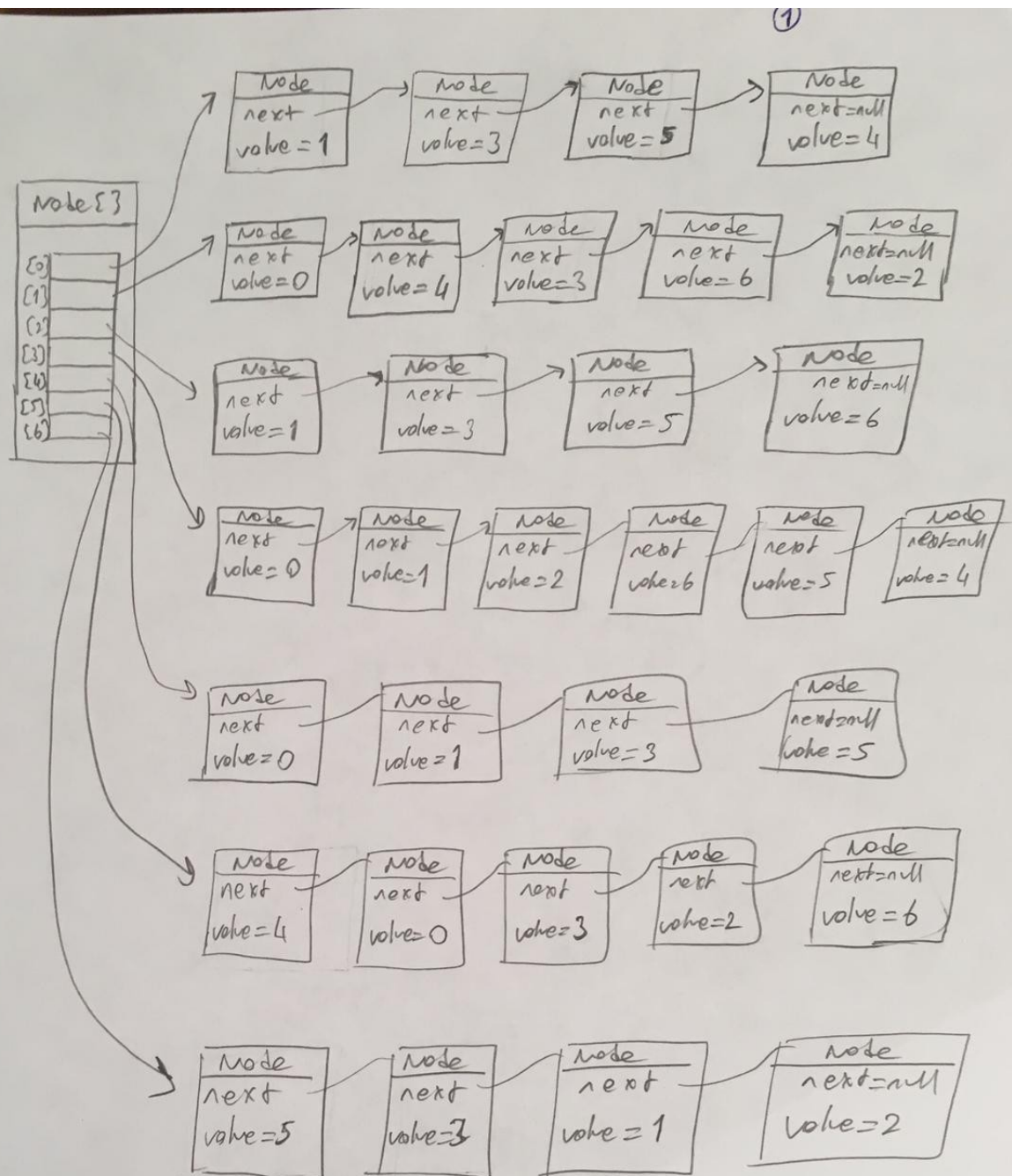
First Graph --> $|V| = 7$, $|E| = 32$, density = $(32/49) = 0.65$

The density more than 0.25. Based on storage efficiency we can conclude that an adjacency matrix representation would be more efficient.

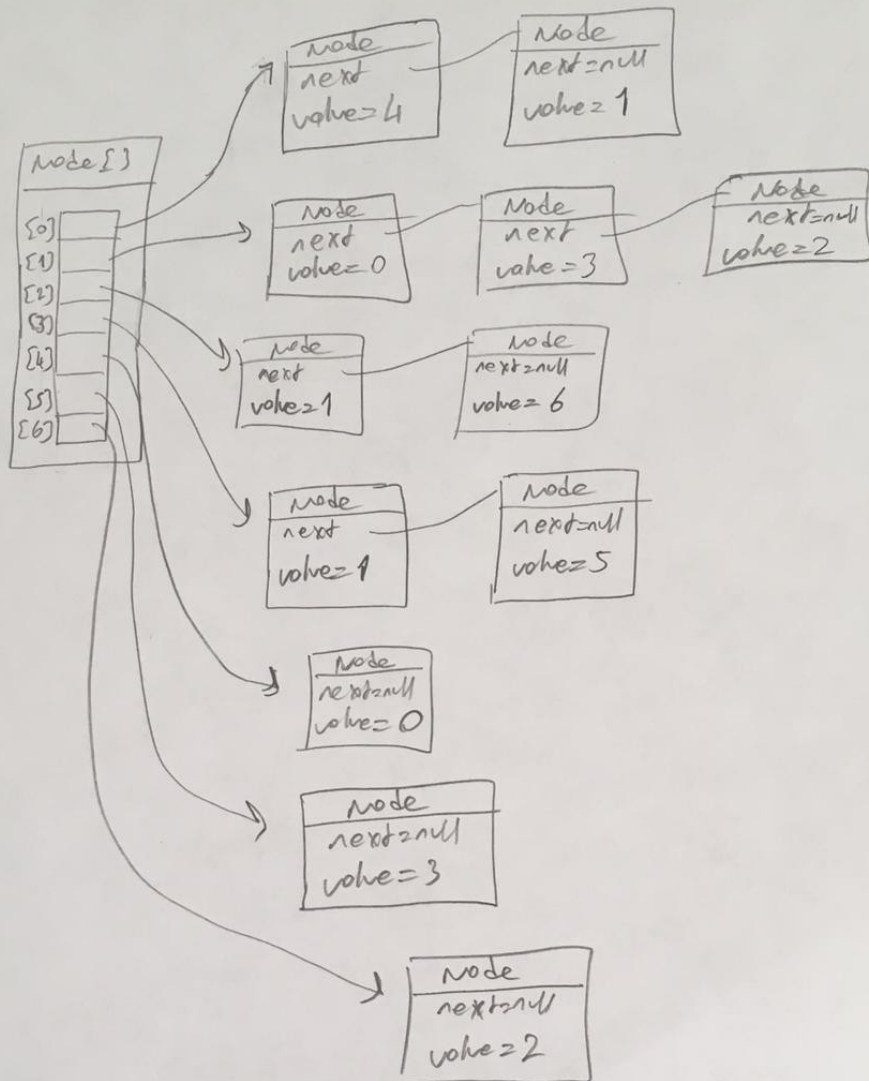
Second Graph -> $|V| = 7$, $|E| = 12$, density = $(12/49) = 0.25$

The density is 0.25, either representation may be appropriate representation. When not rounded, density less than 0.25, suggesting an adjacency list representation may be more appropriate.

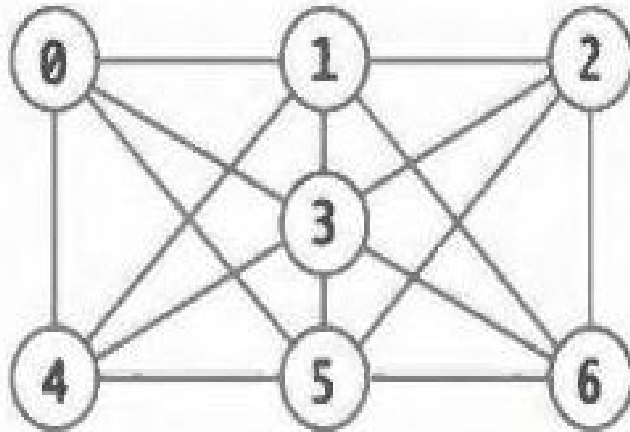
#Represent the graphs above using adjacency lists. Draw the corresponding data structure.



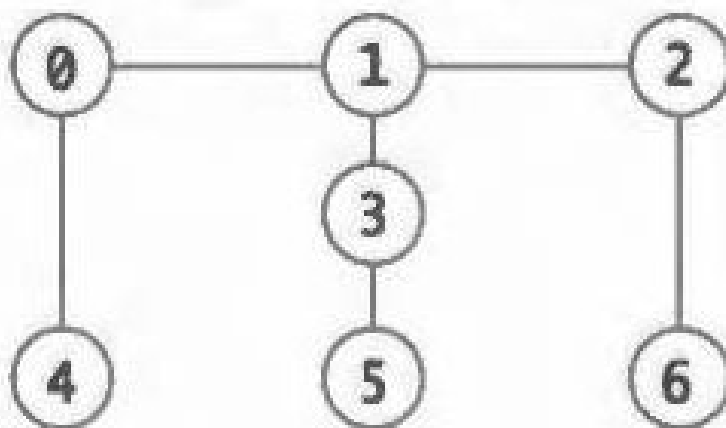
②



#Represent the graphs above using an adjacency matrix. Draw the corresponding data structure.

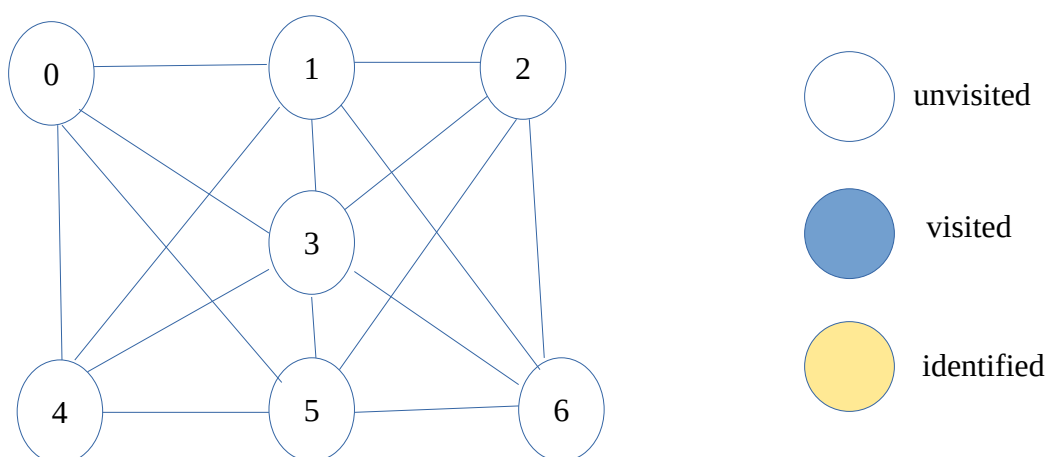


	[0]	[1]	[2]	[3]	[4]	[5]	[6]
[0]		1.0		1.0	1.0	1.0	
[1]	1.0		1.0	1.0	1.0		1.0
[2]		1.0		1.0		1.0	1.0
[3]	1.0	1.0	1.0		1.0	1.0	1.0
[4]	1.0	1.0		1.0		1.0	
[5]	1.0		1.0	1.0	1.0		1.0
[6]		1.0	1.0	1.0		1.0	

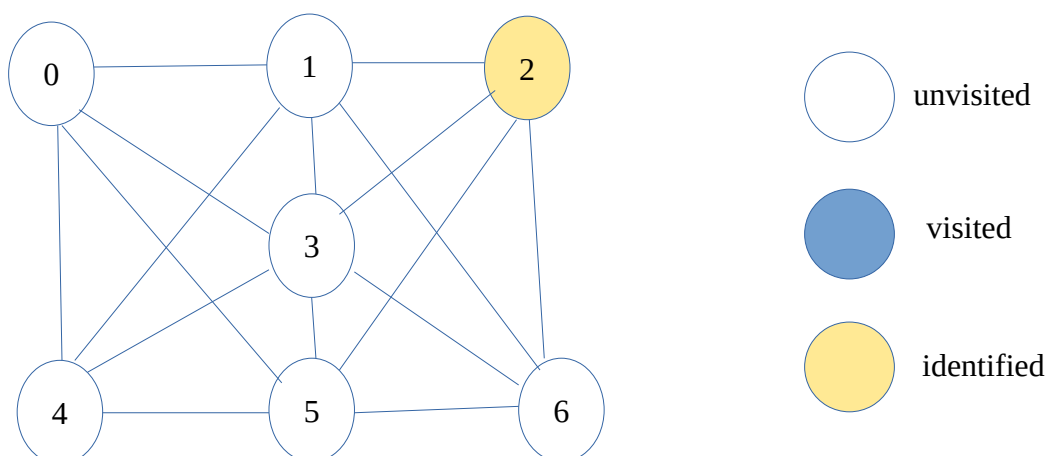


	[0]	[1]	[2]	[3]	[4]	[5]	[6]
[0]		1.0			1.0		
[1]	1.0		1.0	1.0			
[2]		1.0					1.0
[3]		1.0				1.0	
[4]	1.0						
[5]				1.0			
[6]			1.0				

#Draw BFS tree starting from vertex 2 and traversing the vertices adjacent to a vertex in descending order (largest to smallest).

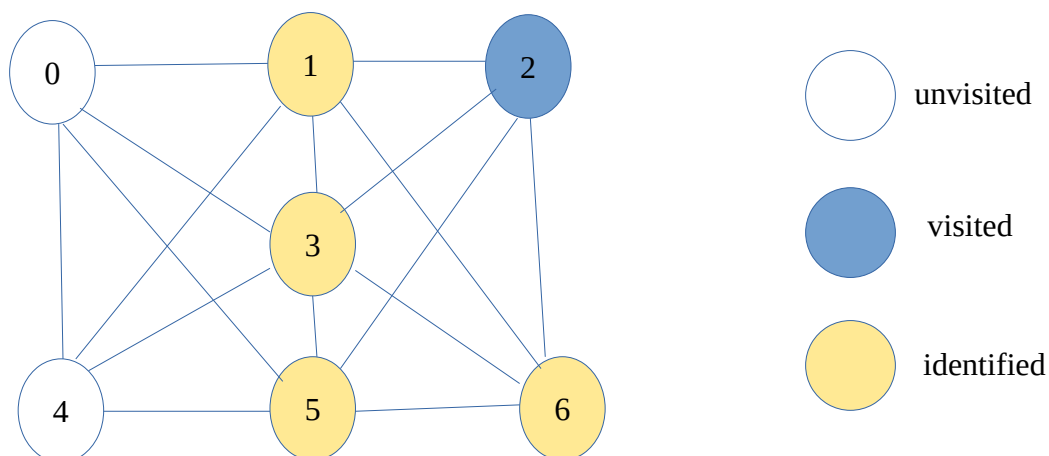
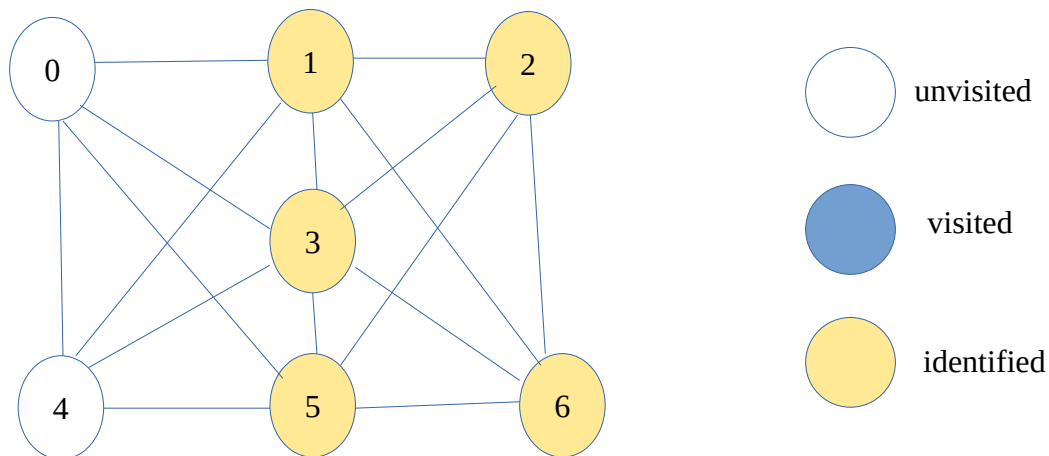


Identity the start node



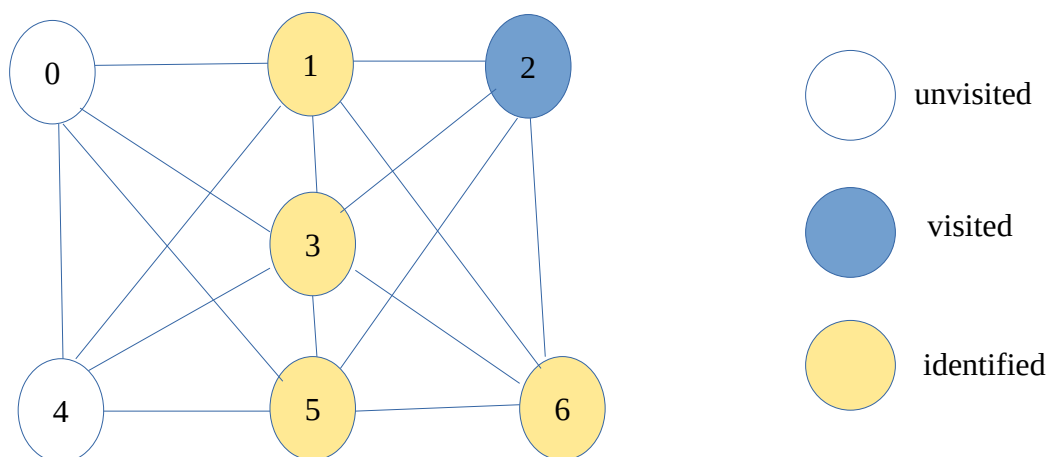
Queue:
6,5,3,1

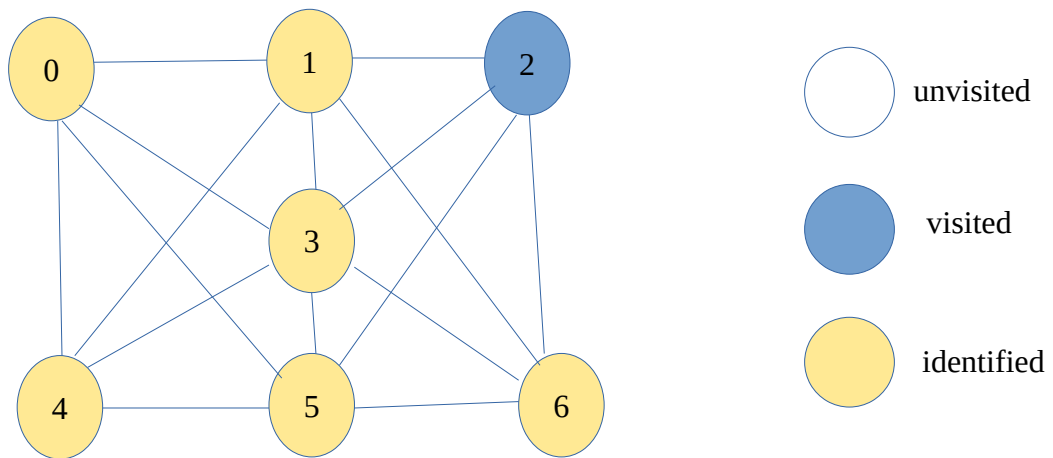
Visit sequence:
2



Queue:
5,3,1

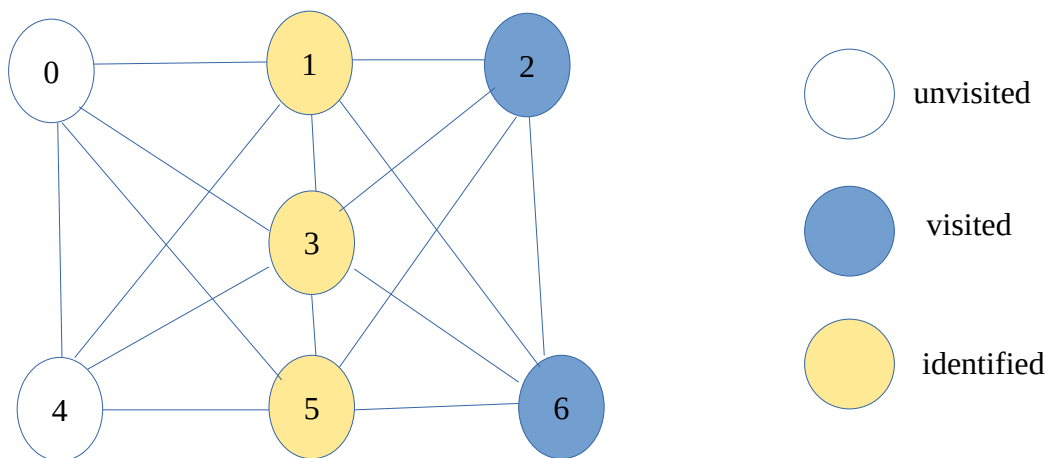
Visit sequence:
2,6





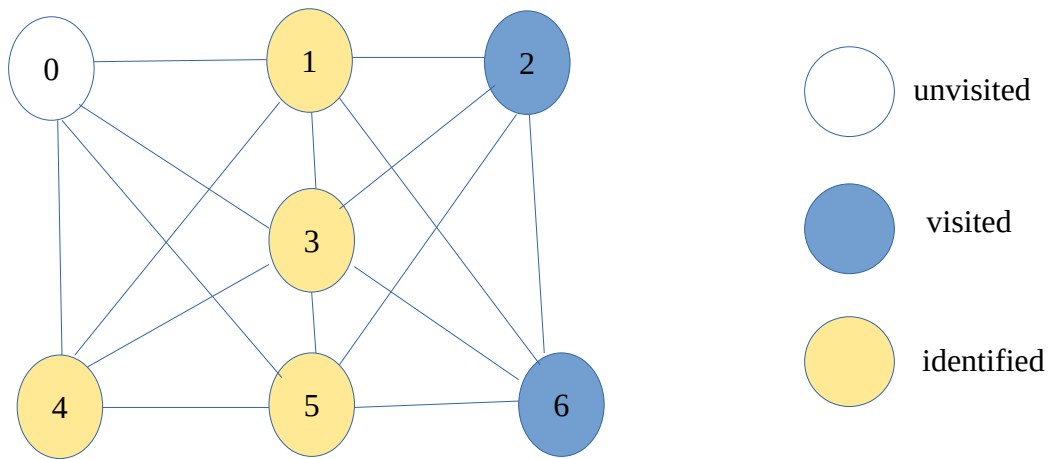
Queue:
5,3,1

Visit sequence:
2,6



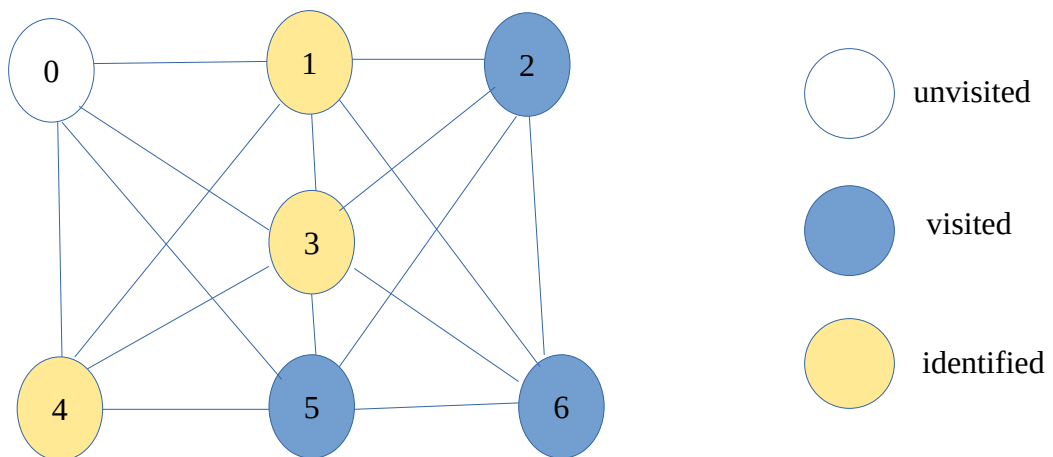
Queue:
3,1

Visit sequence:
2,6,5



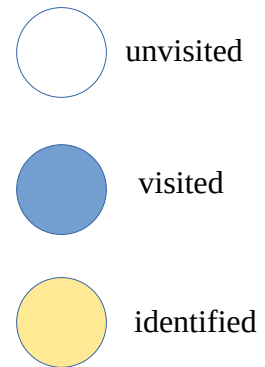
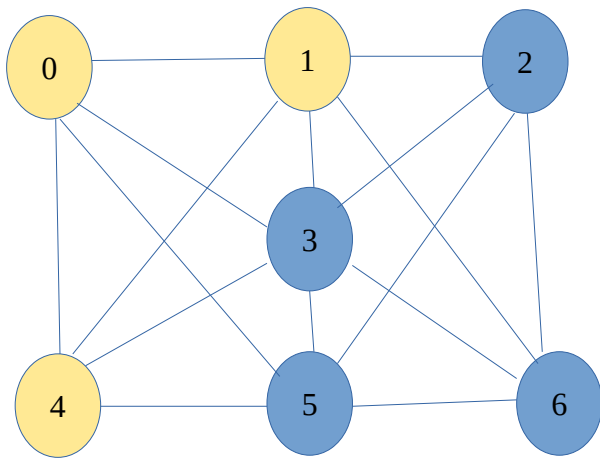
Queue:
3,1,4

Visit sequence:
2,6,5



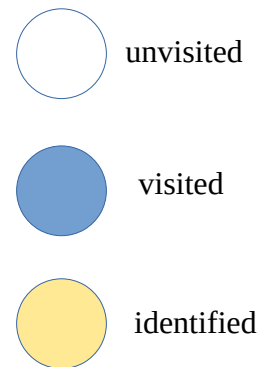
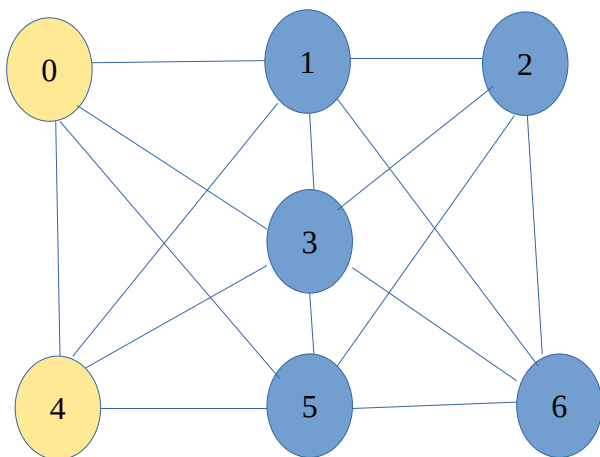
Queue:
3,1,4

Visit sequence:
2,6,5



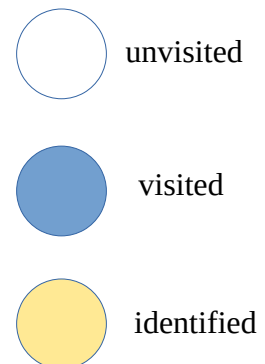
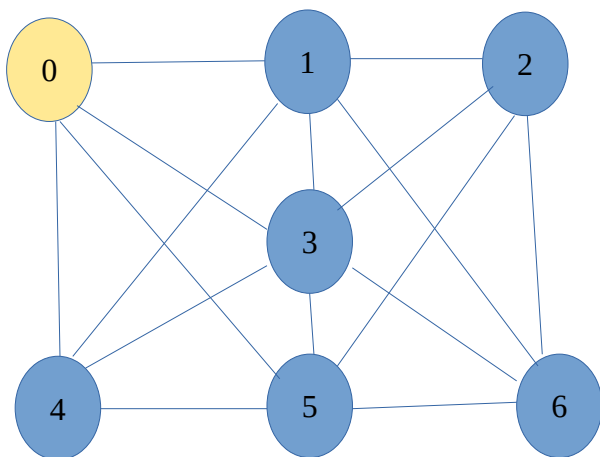
Queue:
1,4,0

Visit sequence:
2,6,5,3



Queue:
4,0

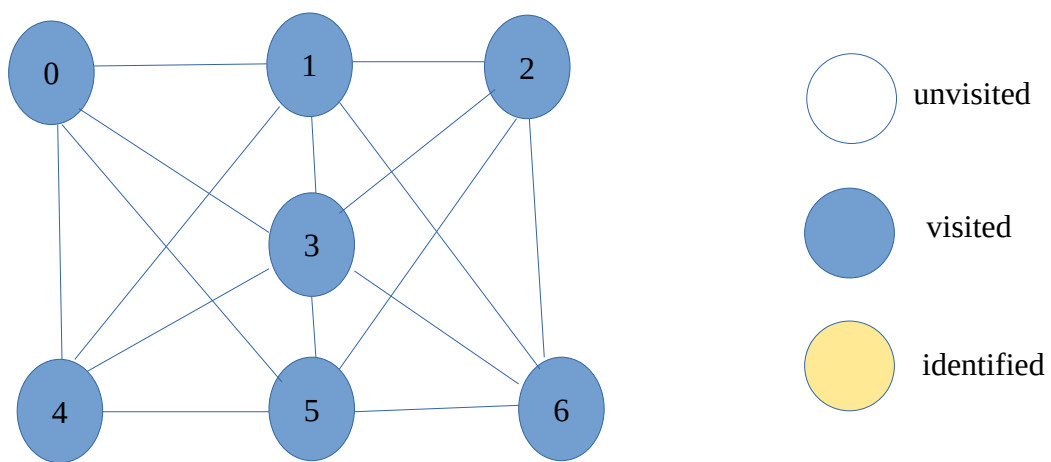
Visit sequence:
2,6,5,3,1



Queue:
0

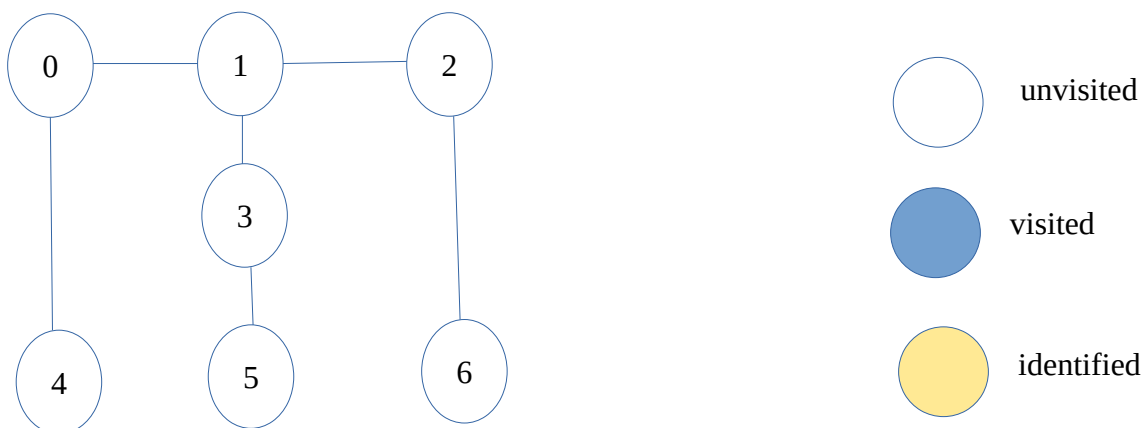
Visit sequence:
2,6,5,3,1,4

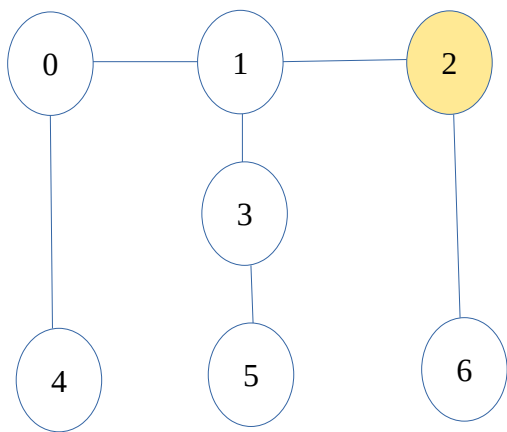
The queue is empty; all vertices have been visited




Queue:
empty


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



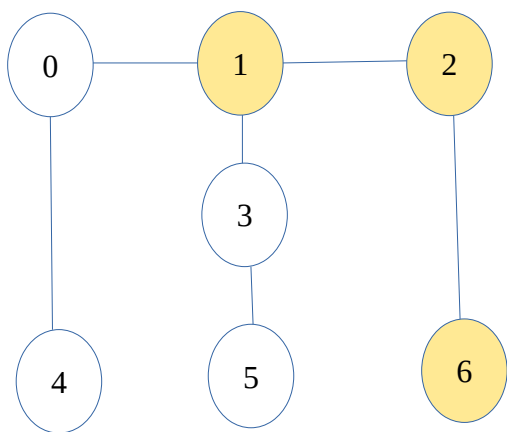



 unvisited

 visited


 identified

Visit sequence:
2



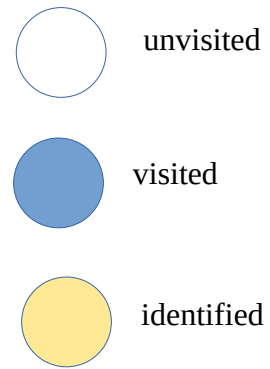
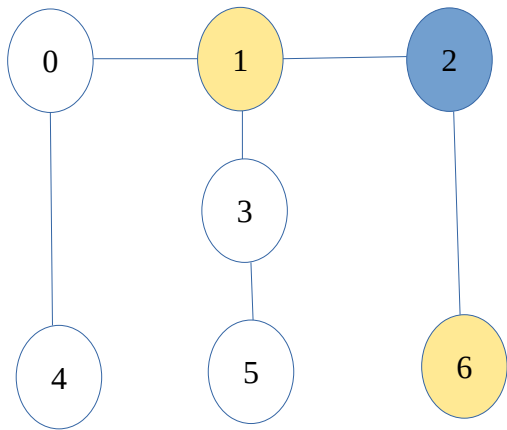
 unvisited

 visited

 identified

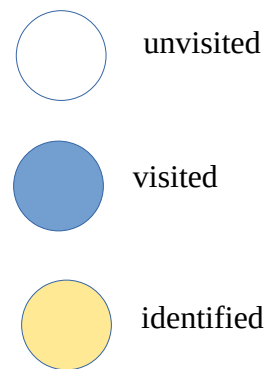
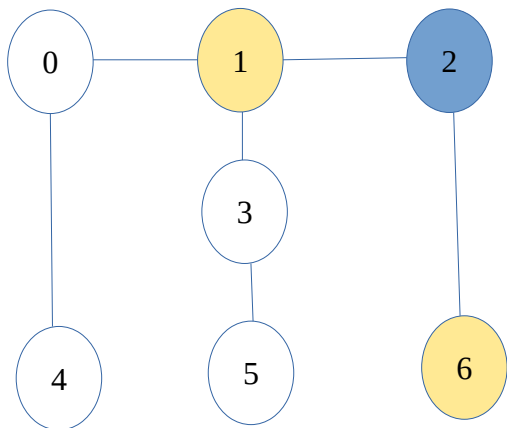
Queue:
6,1

Visit sequence:
2



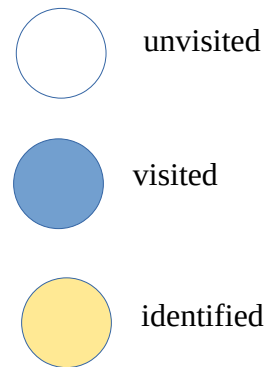
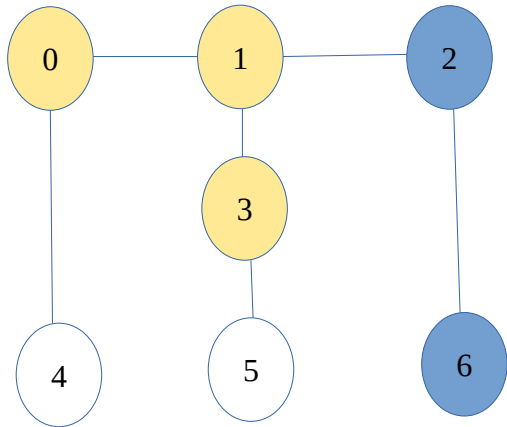
Queue:
6,1

Visit sequence:
2



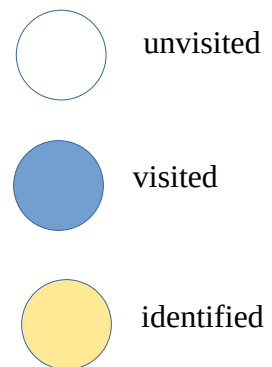
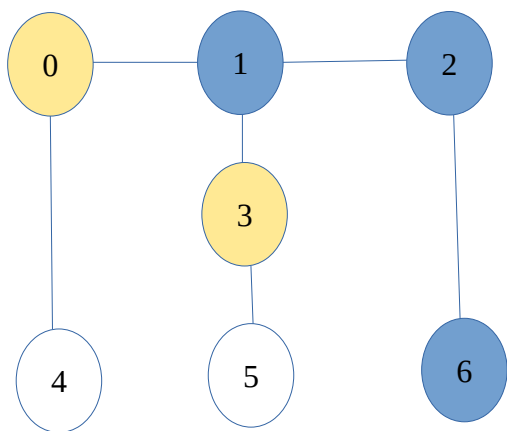
Queue:
1

Visit sequence:
2,6



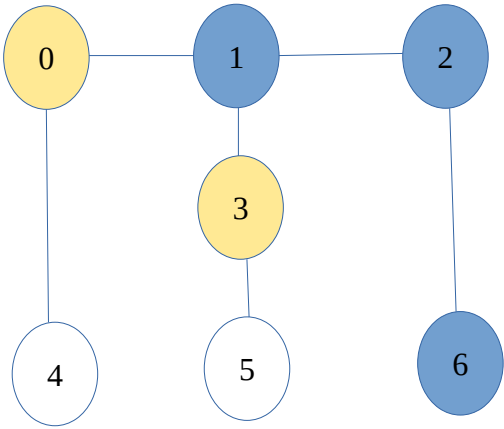
Queue:
3,0


Visit sequence:
2,6,1




Queue:
3,0

Visit sequence:
2,6,1



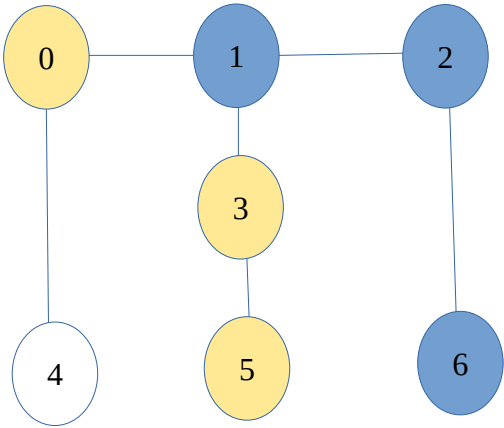
 unvisited


 visited

 identified


Queue:
0

Visit sequence:
2,6,1,3



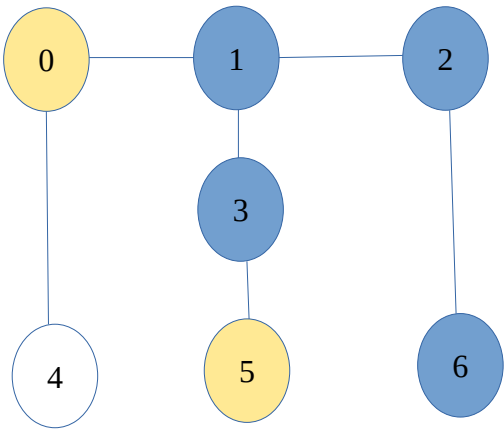
 unvisited


 visited

 identified


Queue:
0,5

Visit sequence:
2,6,1,3



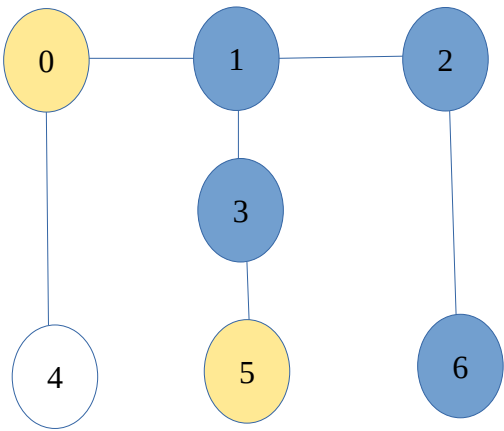
 unvisited


 visited

 identified


Queue:
0,5

Visit sequence:
2,6,1,3



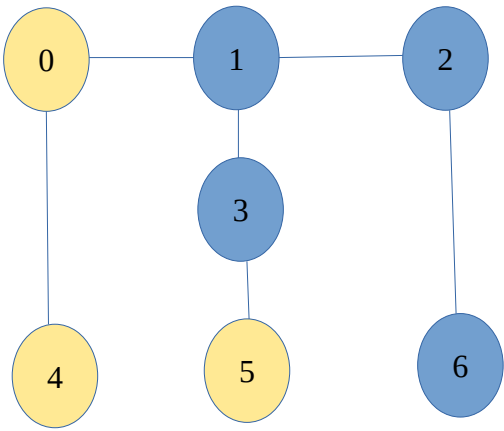
 unvisited


 visited

 identified


Queue:
5

Visit sequence:
2,6,1,3,0



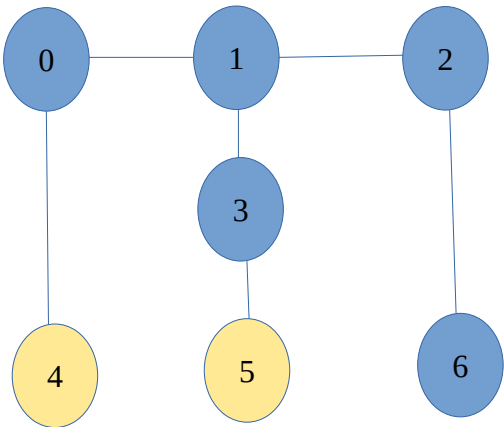
 unvisited


 visited

 identified


Queue:
5,4

Visit sequence:
2,6,1,3,0



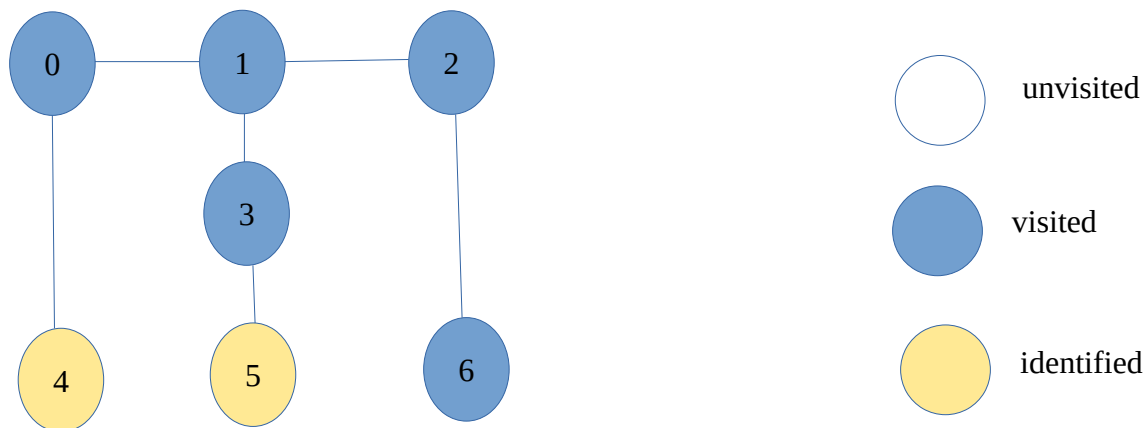
 unvisited

 visited

 identified

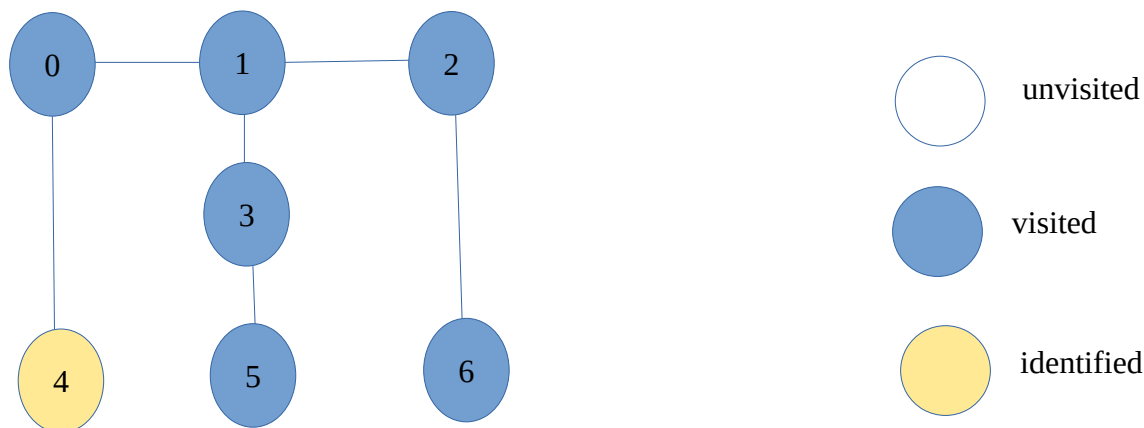
Queue:
5,4

Visit sequence:
2,6,1,3,0



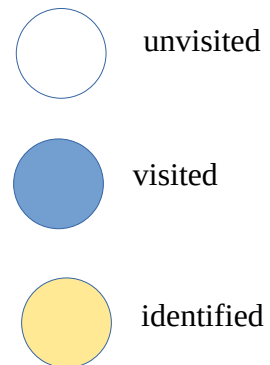
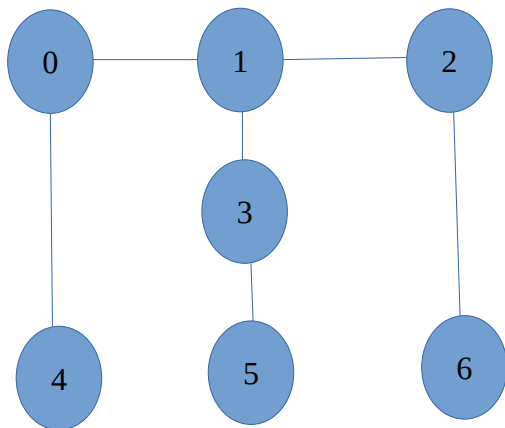
Queue:
4

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



Queue:
4

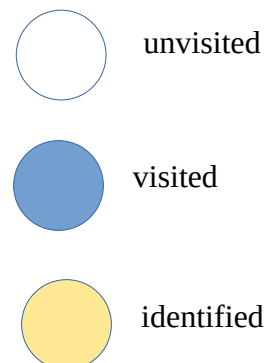
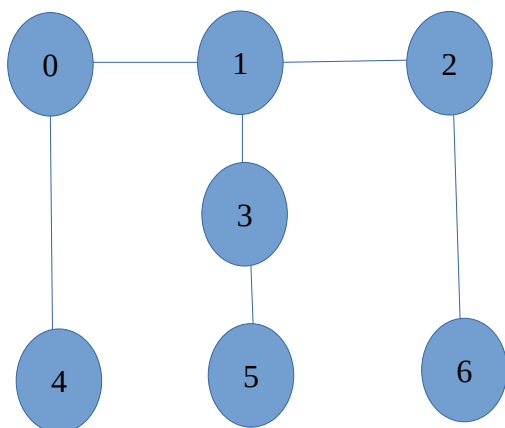
Visit sequence:
2,6,1,3,0,5



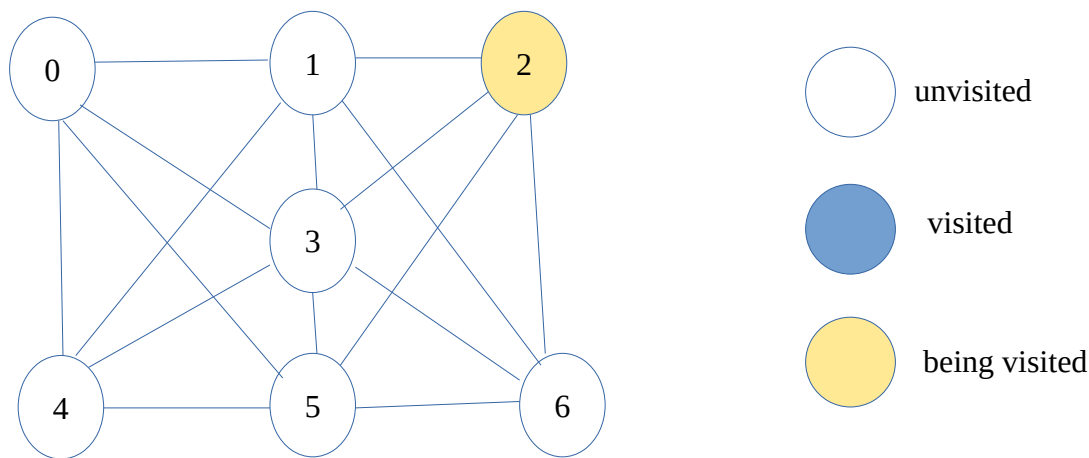
Queue:
empty

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

The queue is empty; all vertices have been visited

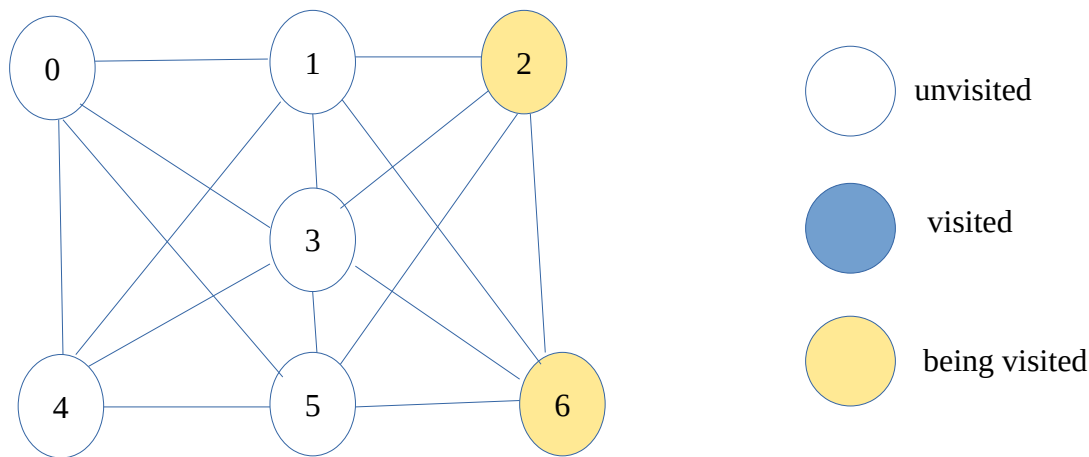


#Draw DFS tree starting from vertex 2 and traversing the vertices adjacent to a vertex in descending order (largest to smallest).



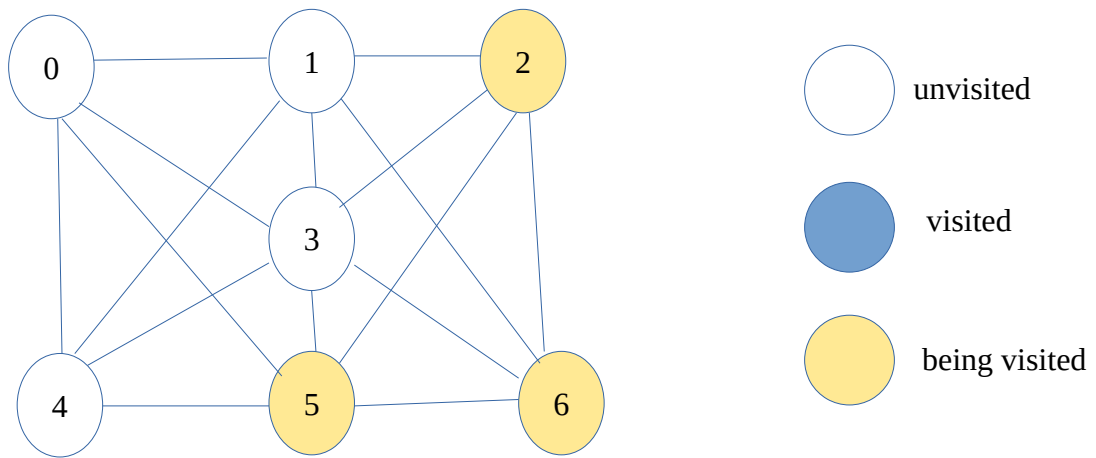
Discovery(visit) order:
2

Finish order:



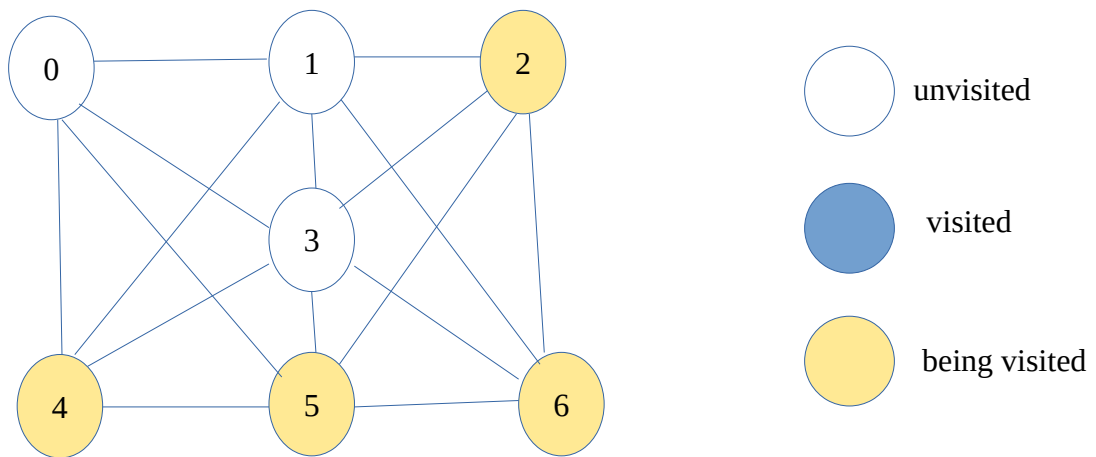
Discovery(visit) order:
2, 6

Finish order:



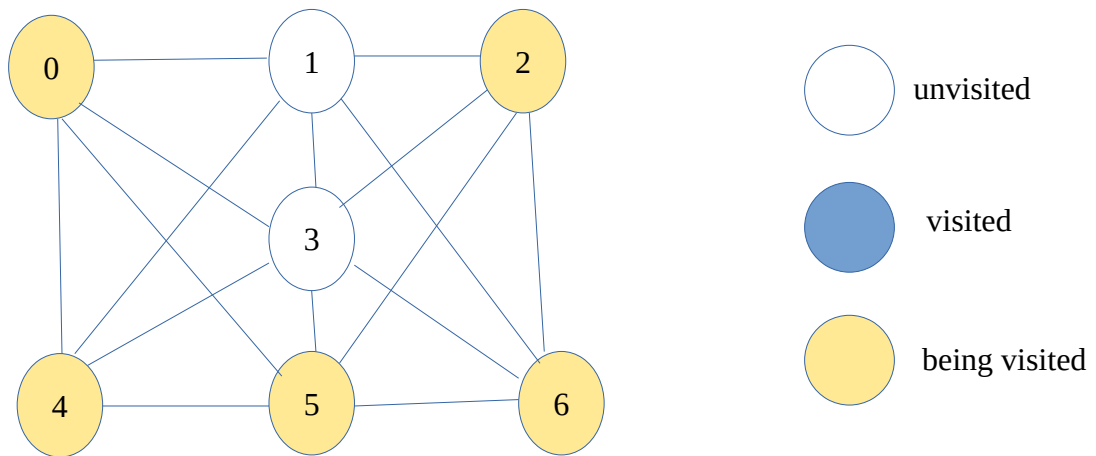
Discovery(visit) order:
2, 6, 5

Finish order:



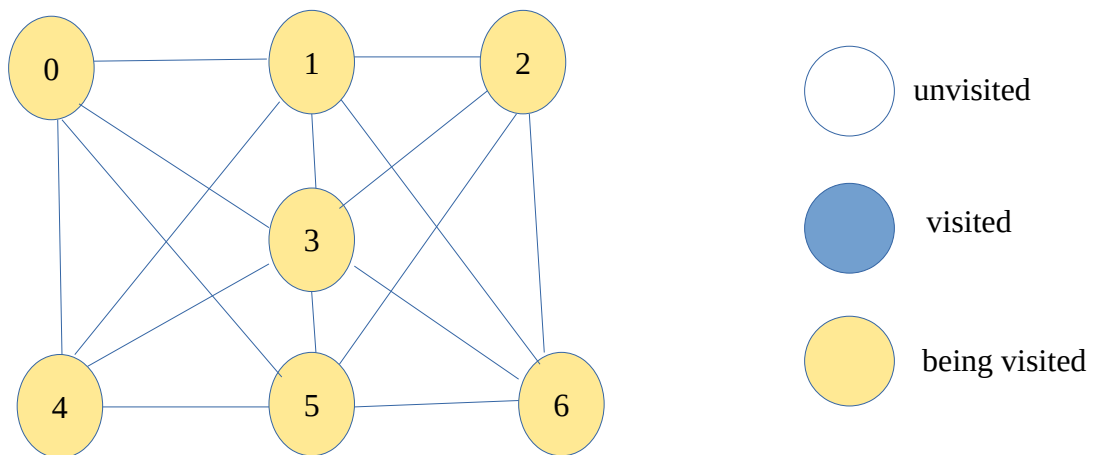
Discovery(visit) order:
2, 6, 5, 4

Finish order:



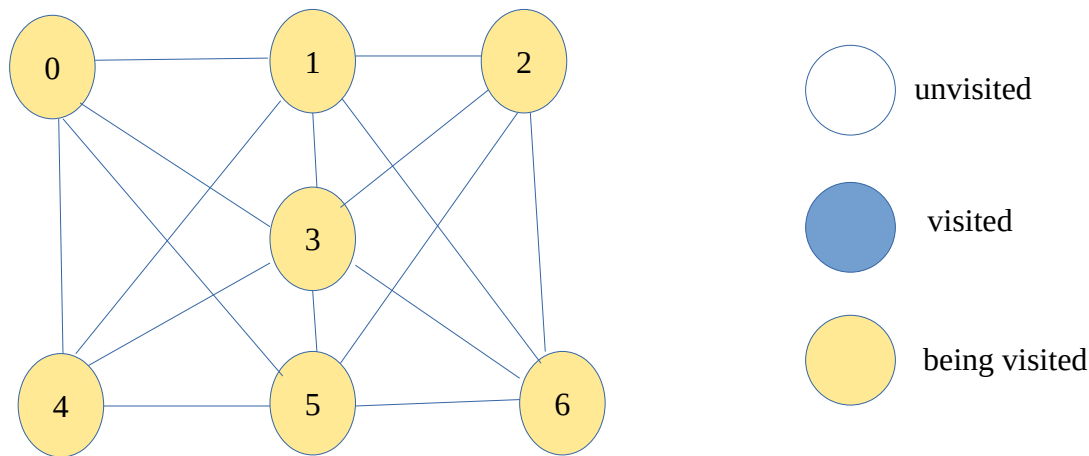
Discovery(visit) order:
2, 6, 5, 4, 0

Finish order:



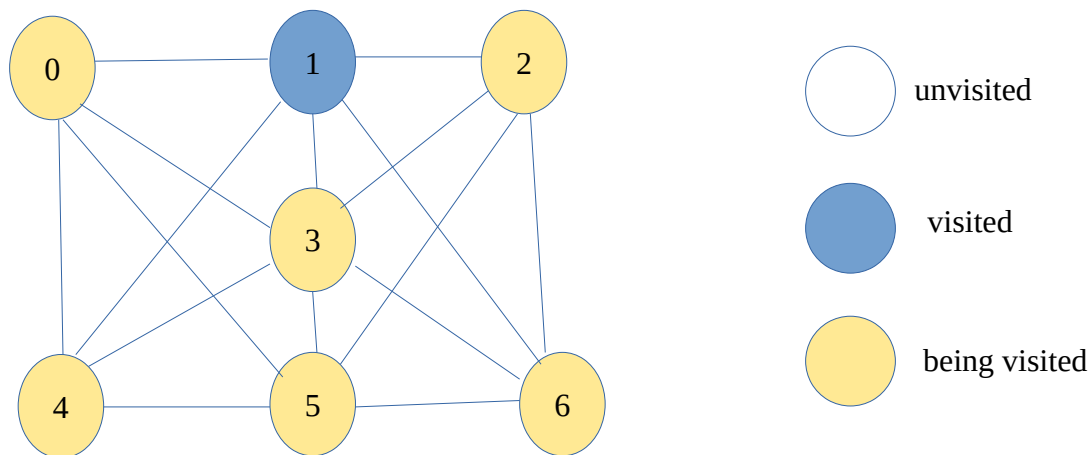
Discovery(visit) order:
2, 6, 5, 4, 0, 3

Finish order:

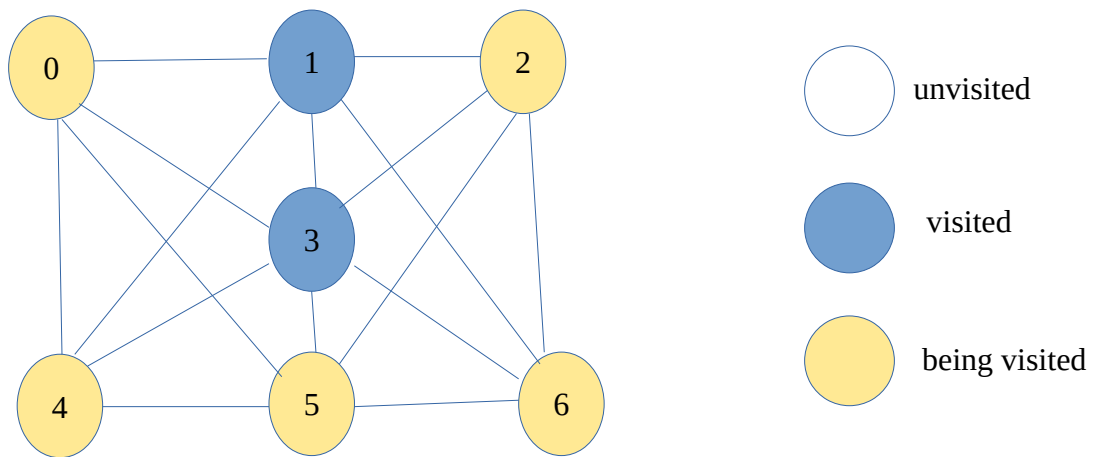


Discovery(visit) order:
2, 6, 5, 4, 0, 3, 1

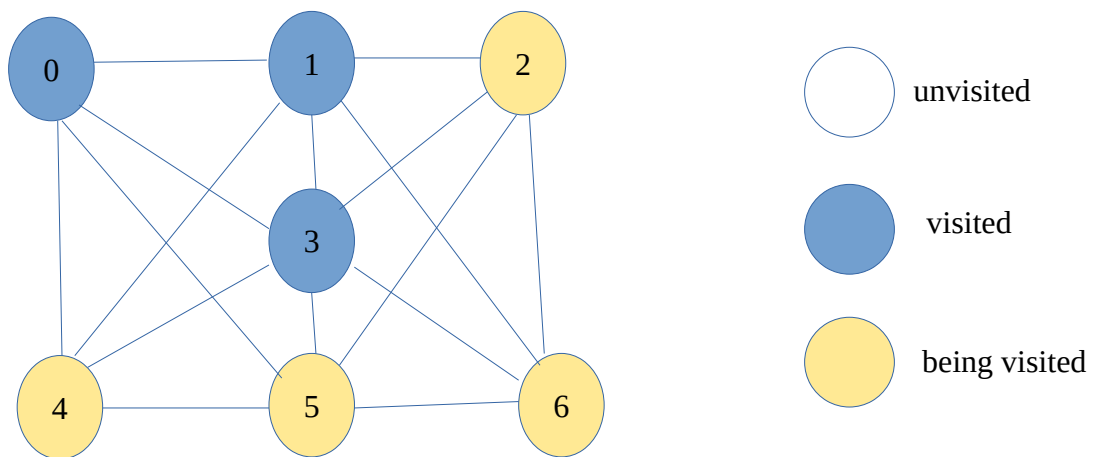
Finish order:



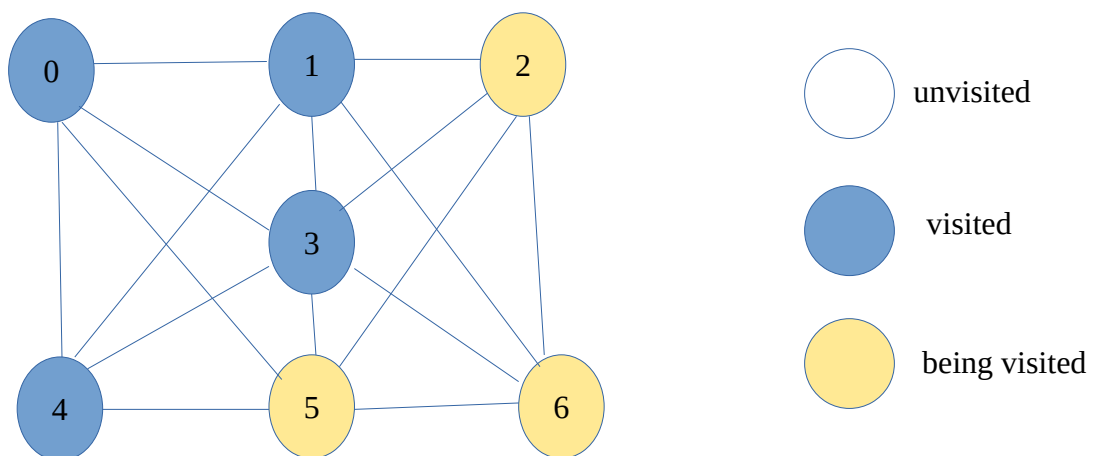
Finish order:
1



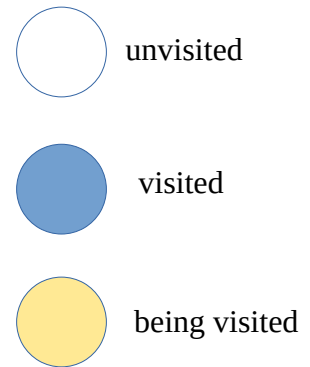
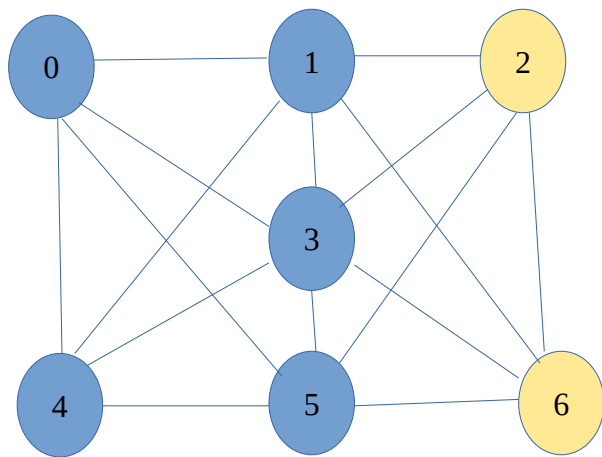
Finish order:
1, 3,



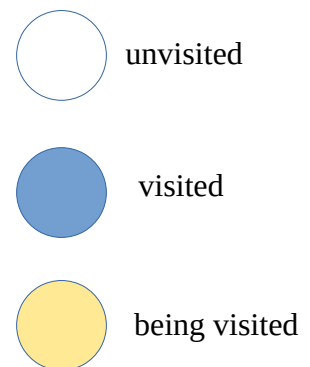
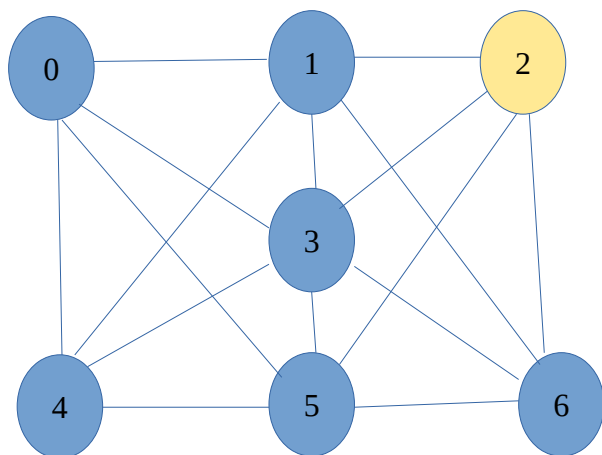
Finish order:
1, 3, 0



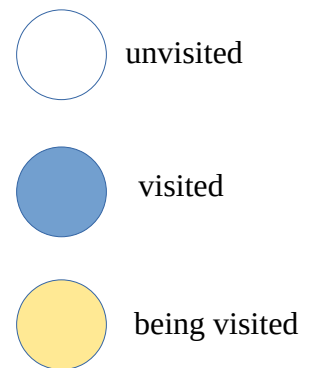
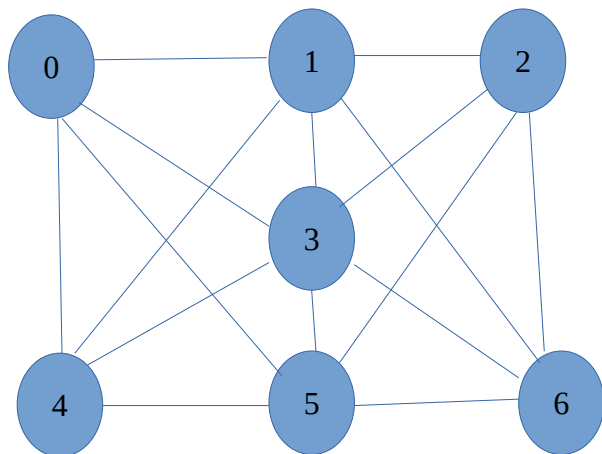
Finish order:
1, 3, 0, 4



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

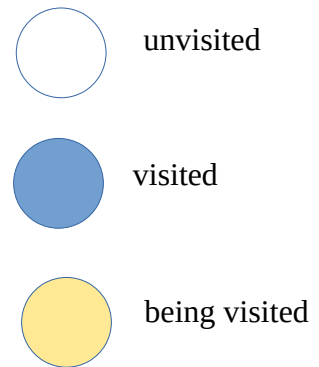
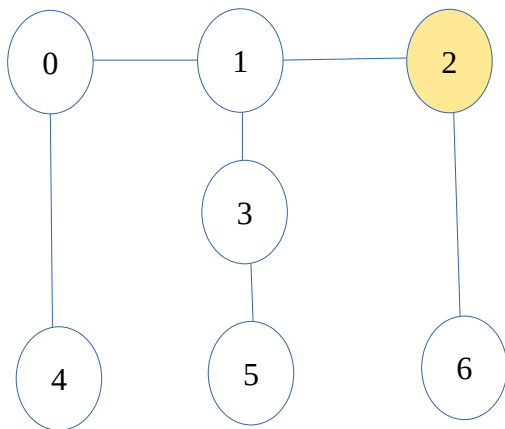


Finish order:
1, 3, 0, 4, 5, 6



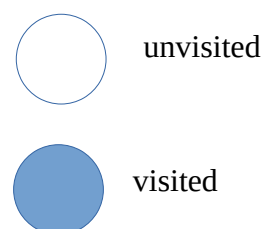
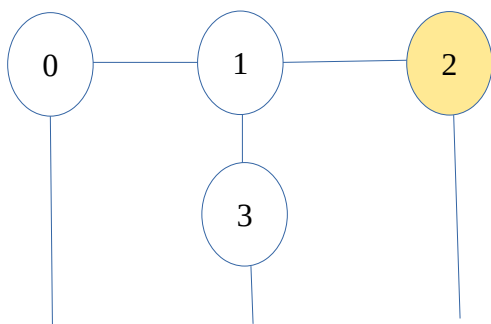
Finish order:

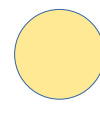
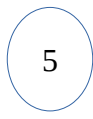
1, 3, 0, 4, 5, 6, 2



Discovery(visit) order:
2

Finish order:

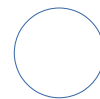
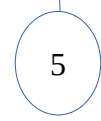
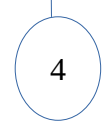
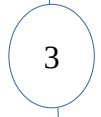
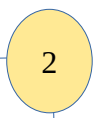
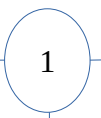




being visited

Discovery(visit) order:
2,6

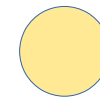
Finish order:



unvisited



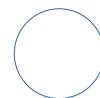
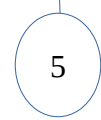
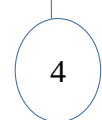
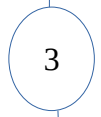
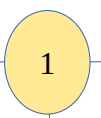
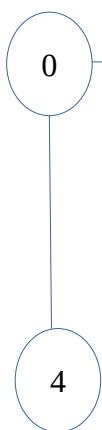
visited



being visited

Discovery(visit) order:
2

Finish order:
6



unvisited



visited



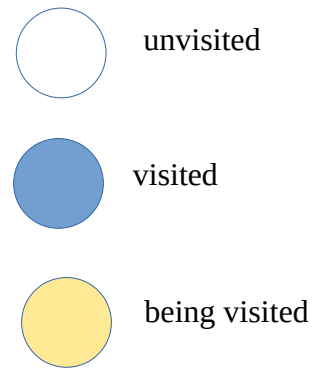
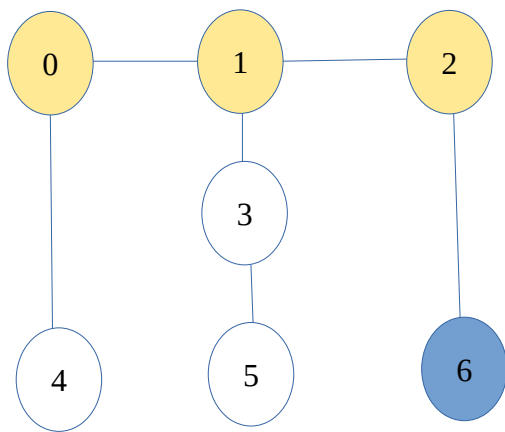
being visited

Discovery(visit) order:

2, 1

Finish order:

6

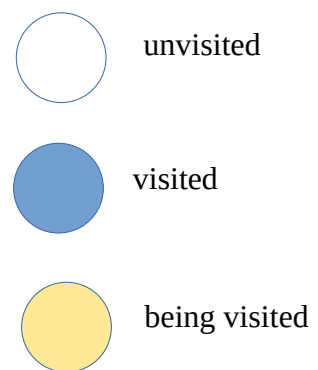
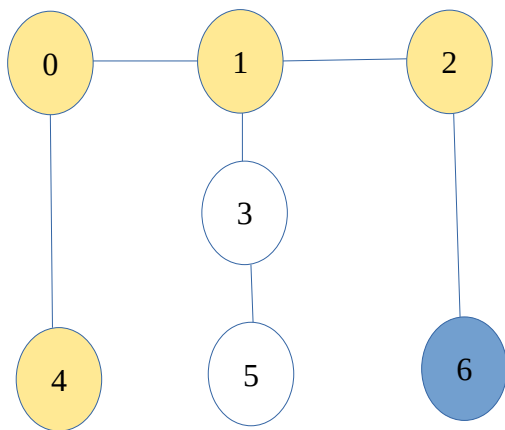


Discovery(visit) order:

2, 1,0

Finish order:

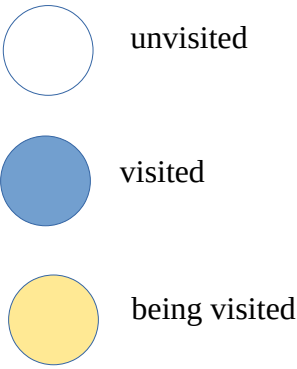
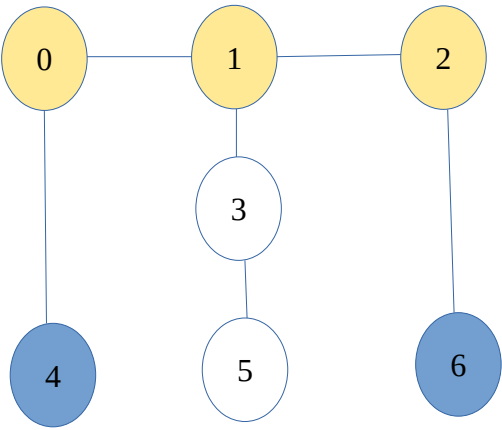
6



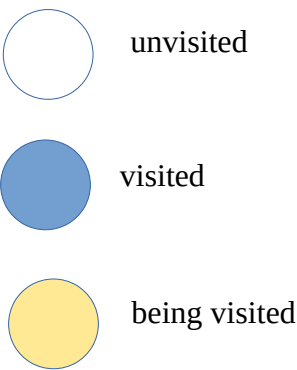
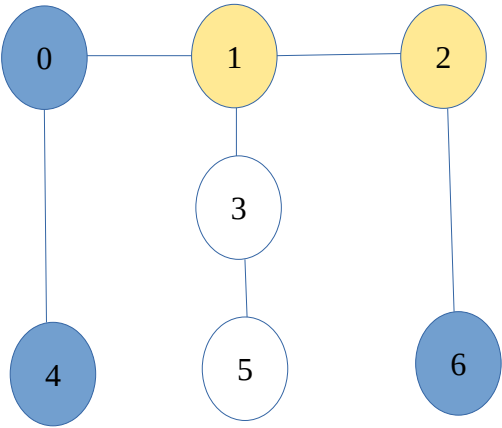
Discovery(visit) order:

2, 1,0,4

Finish order:
6

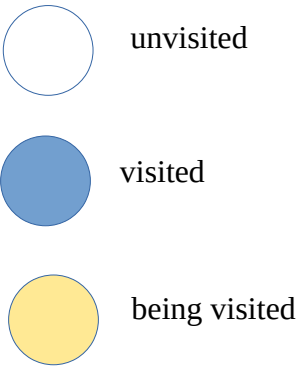
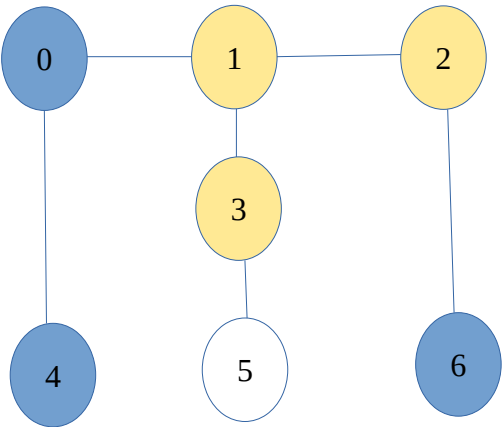


Finish order:
6,4



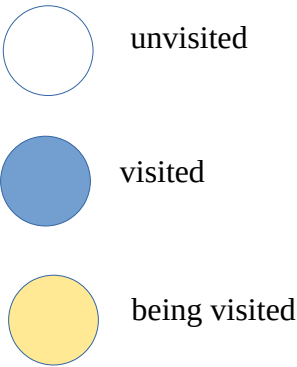
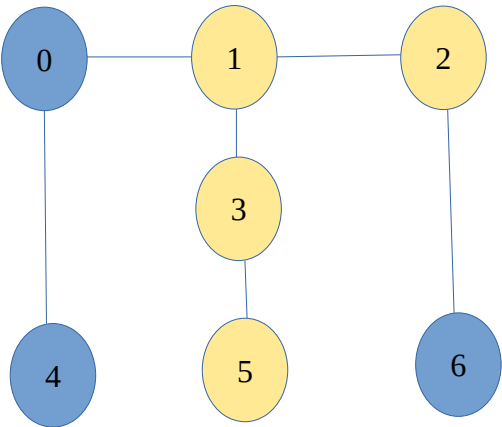
Finish order:

6,4,0



Discovery(visit) order:
2, 1, 3

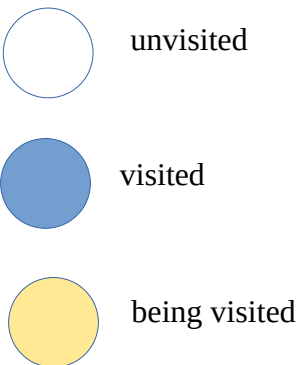
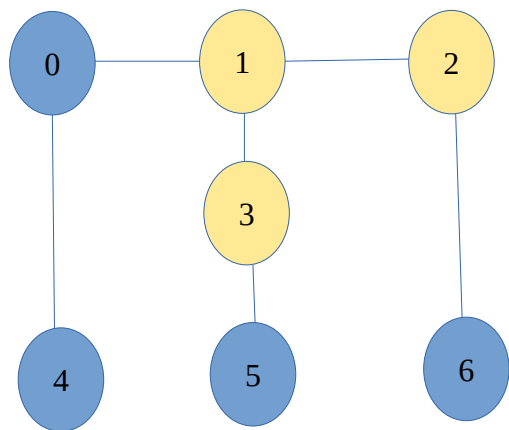
Finish order:
6,4,0



Discovery(visit) order:

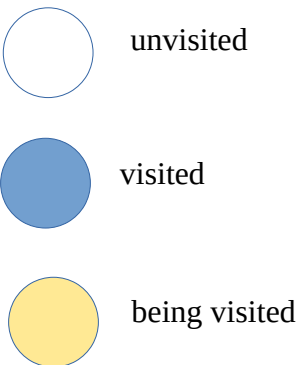
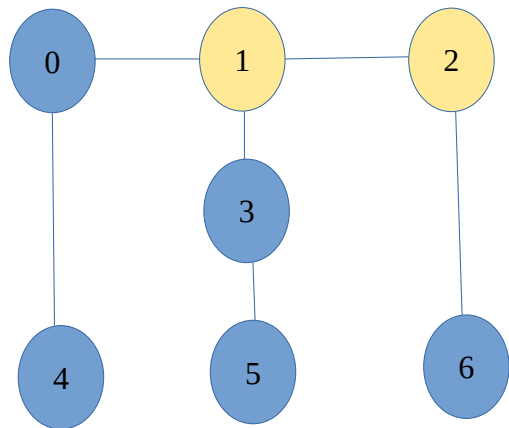
2, 1, 3, 5

Finish order:
6,4,0

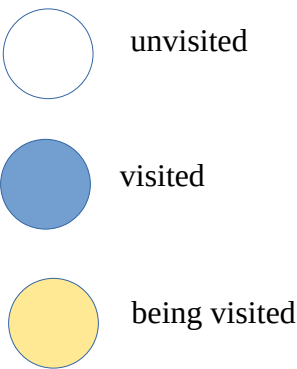
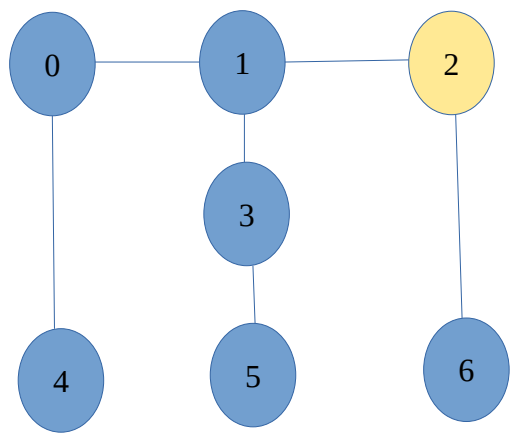


Discovery(visit) order:
2, 1, 3, 5

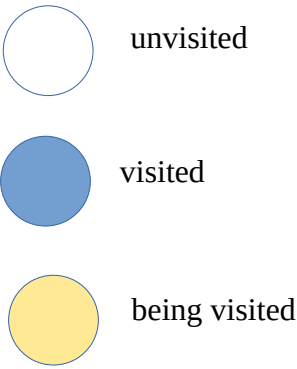
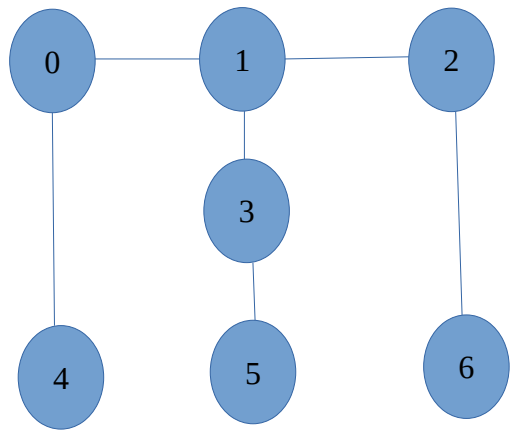
Finish order:
6,4,0,5



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



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



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