

## Assignment 5

Problem statement:

Create an undirected graph using Adjacency Matrix.

The graph must contain 8 nodes and 16 edges.

Do a DFS search on the graph.

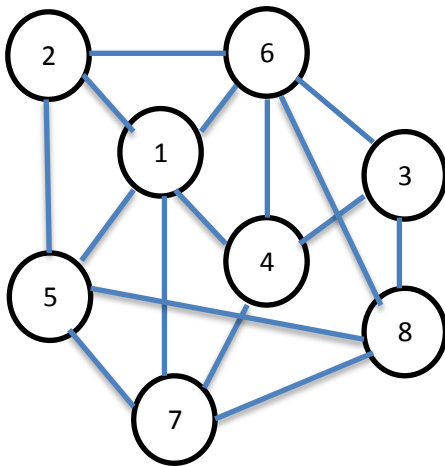
Do a BFS search on the graph.

Your printout will show the adjacency Matrix of the graph

The printout should show the sequence of nodes visited in DFS and BFS.

Solution:

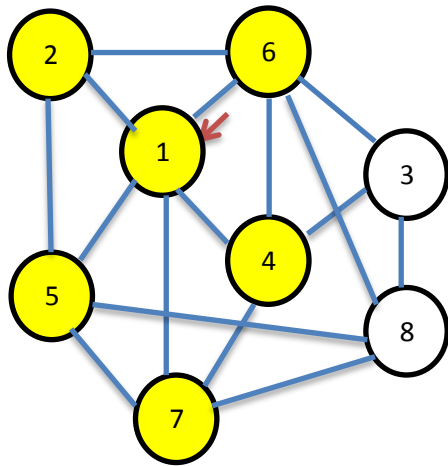
Considering the following graph :



The adjacency matrix representation would look as shown below :

	1	2	3	4	5	6	7	8
1	0	1	0	1	1	1	1	0
2	1	0	0	0	1	1	0	0
3	0	0	0	1	0	1	0	1
4	1	0	1	0	0	1	1	0
5	1	1	0	0	0	0	1	1
6	1	1	1	1	0	0	0	1
7	1	0	0	1	1	0	0	1
8	0	0	1	0	1	1	1	0

Breadth first search :



First iteration :

Start with node 1

Visit Adjacent node 2 and mark as visited. Enqueue 2

Visit adjacent node 4 and mark as visited. Enqueue 4

Visit adjacent node 5 and mark as visited. Enqueue 5

Visit adjacent node 6 and mark as visited. Enqueue 6

Visit adjacent node 7 and mark as visited. Enqueue 7.

Queue:

2

4

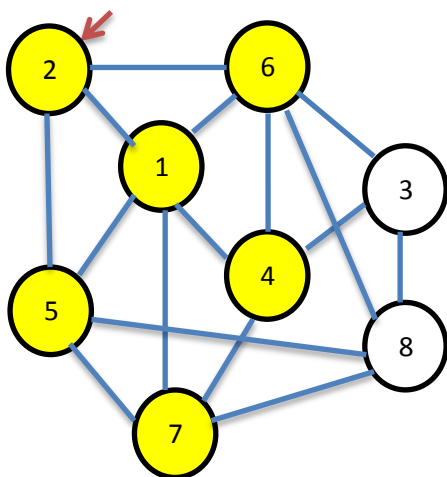
5

6

7

Result:

1,2,4,5,6,7



Second iteration :

Dequeue vertex 2. And start traversal from 2.

Adjacent node 1,5,6 have already been visited.

Queue:

~~2~~

4

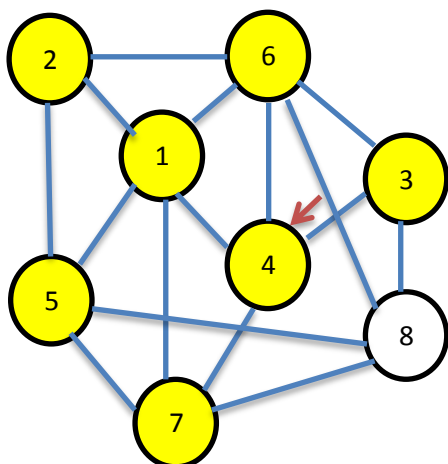
5

6

7

Result:

1,2,4,5,6,7



Third iteration :

Dequeue vertex 4. And start traversal from 4.

Adjacent node 1 and 7 have already been visited.

Visit adjacent node 3 and mark as visited. Enqueue 3.

Queue:

~~2~~

~~4~~

5

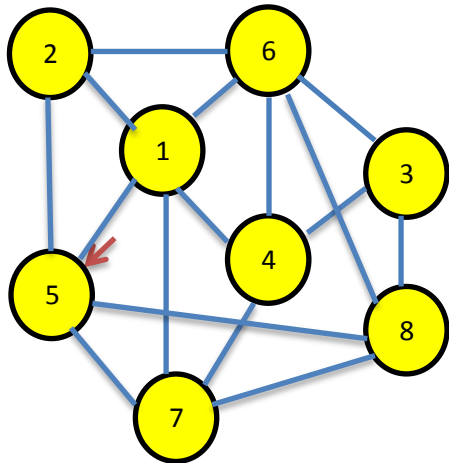
6

7

3

Result:

1,2,4,5,6,7,3



Fourth iteration :

Dequeue vertex 5. And start traversal from 5.

Adjacent node 1,2 and 7 have already been visited.

Visit adjacent node 8 and mark as visited. Enqueue 8

Queue:

~~2~~

~~1~~

~~5~~

6

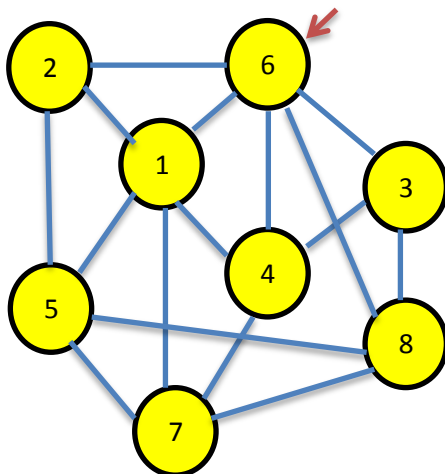
7

3

8

Result:

1,2,4,5,6,7,3,8



Fifth iteration :

Dequeue vertex 6. And start traversal from 6.

All adjacent nodes have been visited

Queue:

~~2~~

~~1~~

~~5~~

~~4~~

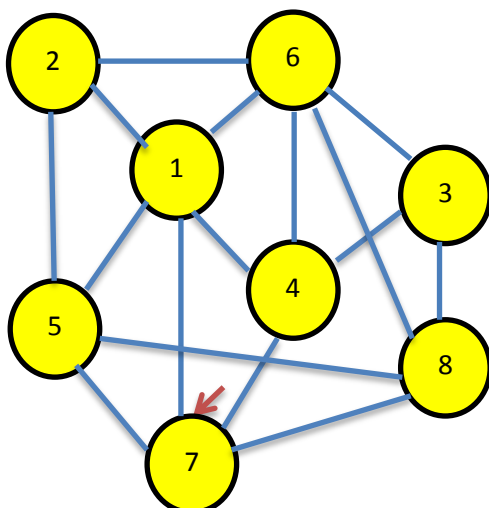
7

3

8

Result:

1,2,4,5,6,7,3,8



Sixth iteration :

Dequeue vertex 7. And start traversal from 7.

All adjacent nodes have been visited.

Queue:

~~2~~

~~1~~

~~5~~

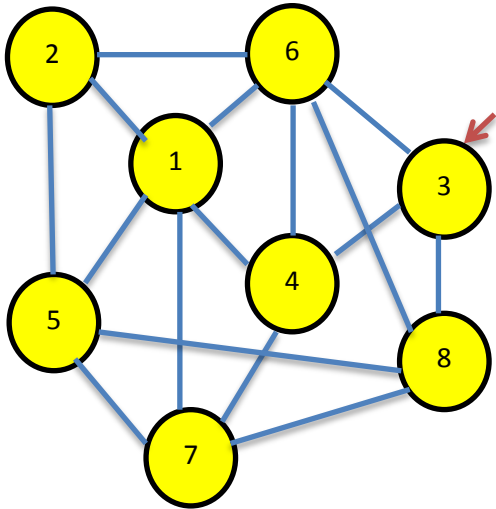
~~4~~

3

8

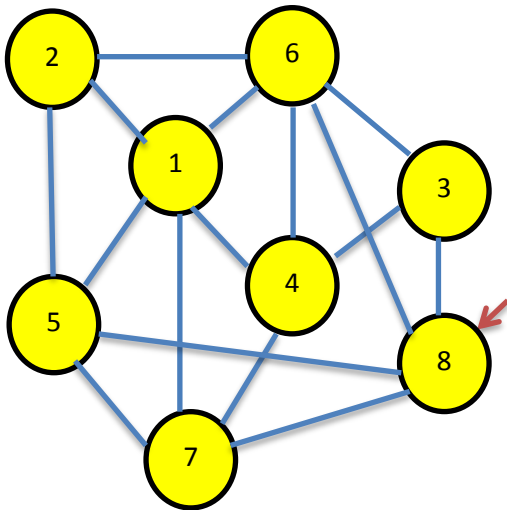
Result:

1,2,4,5,6,7,3,8



Seventh iteration :  
 Dequeue vertex 3. And start traversal  
 from 3.  
 All adjacent nodes have been visited  
 Queue:

~~2~~  
~~4~~  
~~5~~  
~~6~~  
~~7~~  
~~3~~  
 8  
 Result:  
 1,2,4,5,6,7,3,8



Eighth iteration :  
 Dequeue vertex 8. And start traversal  
 from 8.  
 Queue:

~~3~~  
~~4~~  
~~5~~  
~~6~~  
~~7~~  
~~8~~  
 All adjacent nodes have been visited and  
 the queue is empty. Hence the result is

Result:  
 1,2,4,5,6,7,3,8

Depth first search :

In a similar fashion, depth first search uses stack implementation instead of queues to traverse the graph.

The result for depth first search in the above case would be :

1,2,5,7,4,3,6,8