

Graph Traversal:

- The graph traversal is used to decide the order of vertices visited in the search process.

i.e. using graph traversal all the vertices of the graph are visited without getting into loops.

- There are two graph traversal techniques used:

1. DFS (Depth First Search)
2. BFS (Breadth First Search)

1. DFS Traversal: (uses stack)

- DFS algorithm starts with an initial node of the graph G, and then goes to deeper and deeper until the searching node or the node which has no neighbour is found.

- The algorithm, then backtracks from the dead end towards the most recent node that is yet to be completely unexplored.

- The Stack data structure is used to implement DFS traversal

- The maximum size of stack used is the total number of vertices in the graph.

Algorithm:

Step 1: Initialize all nodes in graph G to ready state

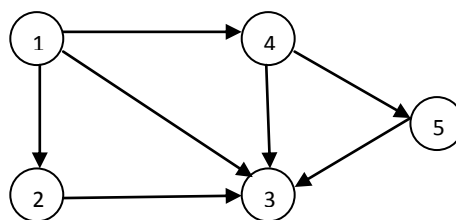
Step 2: Push the starting node on to the stack and change its state to waiting state

Step 3: Repeat Steps 4 and 5 until stack is empty

Step 4: Pop the top node N from stack, process it and change its state to processed state

Step 5: Push all neighbours of processed node N that are in the ready state on to the stack and change their state to waiting state

Example:



G

Node	Adjacent list
1	2, 3, 4
2	3
3	-
4	3, 5
5	3

Maximum stack size= 5

Initially stack is empty

Initialize all vertices to ready state and select the starting state.

Let the starting vertex be vertex 1.

The sequences of steps executed using DFS algorithm are:

<u>Stack</u>	<u>Ready State</u>	<u>Waiting State</u>	<u>Processed State</u>			
<table><tr><td></td></tr><tr><td></td></tr></table>			1, 2, 3, 4, 5	-	-	
<table><tr><td></td></tr><tr><td>1</td></tr></table>		1	2, 3, 4, 5	1	-	
1						
<table><tr><td>4</td></tr><tr><td>3</td></tr><tr><td>2</td></tr></table>	4	3	2	5	2, 3, 4	1
4						
3						
2						
<table><tr><td>5</td></tr><tr><td>3</td></tr><tr><td>2</td></tr></table>	5	3	2	-	2, 3, 5	1, 4
5						
3						
2						
<table><tr><td></td></tr><tr><td>3</td></tr><tr><td>2</td></tr></table>		3	2	-	2, 3	1, 4, 5
3						
2						
<table><tr><td></td></tr><tr><td></td></tr><tr><td>2</td></tr></table>			2	-	2	1, 4, 5, 3
2						
<table><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr></table>				-	-	1, 4, 5, 3, 2

The stack is now empty.

So DFS traversal of graph G starting at node 1 is complete.

The nodes are processed in the order: **1, 4, 5, 3, 2**.

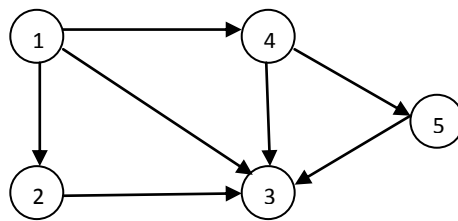
2. BFS Traversal: (*uses Queue*)

- It starts from the starting node and explores all of the neighbours at present depth before moving to next depth level.
- The queue data structure is used to implement BFS traversal
- The maximum size of queue used is the total number of vertices in the graph.

Algorithm:

- Step 1: Initialize all nodes in graph G to ready state
 Step 2: Insert the starting node in to the queue and change its state to waiting state
 Step 3: Repeat Steps 4 and 5 until queue is empty
 Step 4: Remove the front node N from queue, process it and change its state to processed state
 Step 5: Insert all neighbours of processed node N that are in the ready state in to the queue and change their state to waiting state

Example:



Node	Adjacent list
1	2, 3, 4
2	3
3	-
4	3, 5
5	3

Maximum queue size= 5

Initially queue is empty

Initialize all vertices to ready state.

Let the starting state be vertex 1.

The sequences of steps executed using BFS algorithm are:

<u>Queue</u>	<u>Ready State</u>	<u>Waiting State</u>	<u>Processed State</u>
<div><div></div><div></div><div></div><div></div><div></div></div>	1, 2, 3, 4, 5	-	-
<div><div>1</div><div></div><div></div><div></div><div></div></div>	2, 3, 4, 5	1	-
<div><div>2</div><div>3</div><div>4</div><div></div><div></div></div>	5	2, 3, 4	1
<div><div>3</div><div>4</div><div></div><div></div></div>	5	3, 4	1, 2
<div><div>4</div><div></div><div></div><div></div><div></div></div>	5	4	1, 2, 3
<div><div>5</div><div></div><div></div><div></div><div></div></div>	-	5	1, 2, 3, 4
<div><div></div><div></div><div></div><div></div><div></div></div>	-	-	1, 2, 3, 4, 5

The queue is now empty. So BFS Traversal of graph G starting at node 1 is complete.

The nodes are processed in the order: **1, 2, 3, 4, 5.**

Questions to practice:

1. Find the sequence of steps executed and the order in which nodes are processed using DFS and BFS on above graph with starting vertex 2.
2. Find the sequence of steps executed and the order in which nodes are processed using DFS and BFS on below given graph.

