

Graph Traversing

Breadth First Search

The general idea behind breadth first traversal is that, start at a random vertex, then visit all of its neighbors, the first vertex that we visit, say is at level '0' and the neighbors are at level '1'.

After visiting all the vertices at level '1' we then pick one of these vertexes at level '1' and visit all its unvisited neighbors, we repeat this procedure for all other vertices at level '1'.

Say neighbors of level 1 are in level 2, now we will visit the neighbors of all the vertices at level 2, and this procedure will continue.

Also referred as **level ordering** or **level traversing**.

BFS Algorithm

- Step1. Initialize all the vertices to ready state (STATUS = 1)
- Step2. Put the starting vertex into QUEUE and change its status to waiting (STATUS = 2)
- Step 3: Repeat Step 4 and 5 until QUEUE is EMPTY
- Step 4: Remove the front vertex from QUEUE, Process the vertex, Change its status to processed state (STATUS = 3)
- Step 5: ADD all the neighbors in the ready state (STATUS = 1) to the RARE of the QUEUE and change their status to waiting state (STATUS = 2)
- Step 6: Exit.

Depth First Search

The general idea behind depth first traversal is that, starting from any random vertex, single path is traversed until a vertex is found whose all the neighbors are already been visited.

The search then backtracks on the path until a vertex with unvisited adjacent vertices is found and then begin traversing a new path starting from that vertex, and so on.

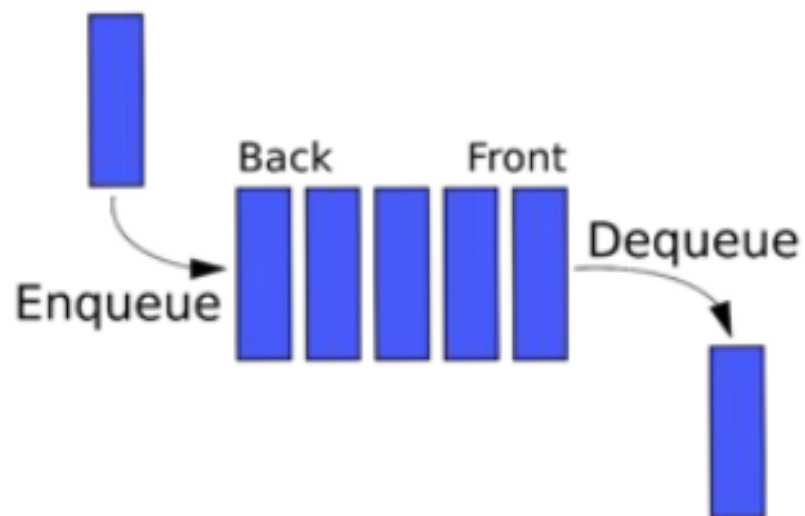
This process will continue until all the vertices of the graph are visited.

DFS Algorithm

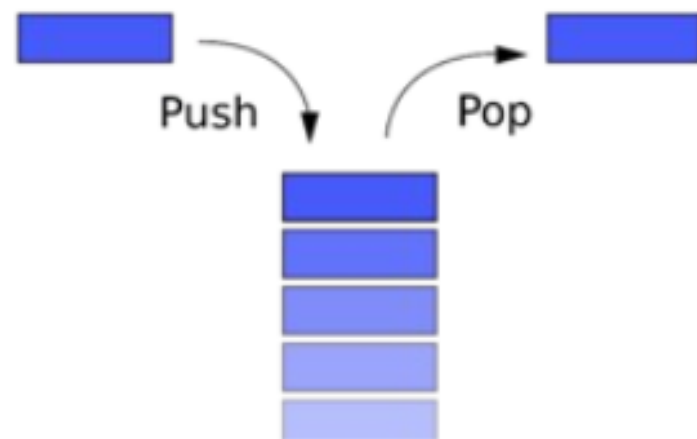
- Step1. Initialize all the vertices to ready state (STATUS = 1)
- Step2. Put the starting vertex into STACK and change its status to waiting (STATUS = 2)
- Step 3: Repeat Step 4 and 5 until STACK is EMPTY
- Step 4: POP the top vertex from STACK, Process the vertex, Change its status to processed state (STATUS = 3)
- Step 5: PUSH all the neighbors in the ready state (STATUS = 1) to the STACK and change their status to waiting state (STATUS = 2)
- Step 6: Exit.

Breadth First Search vs. Depth First Search

BFS uses "first in first out".

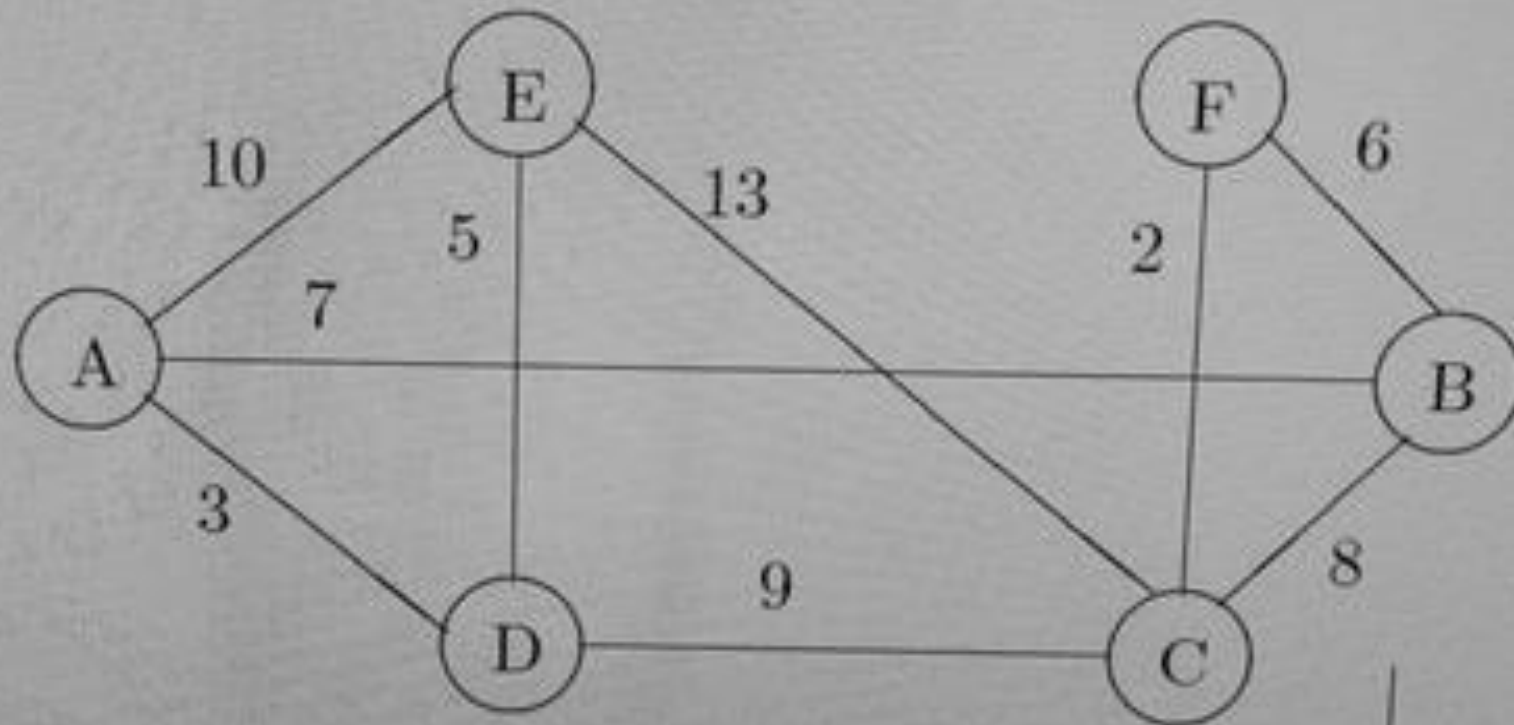


DFS uses "last in first out".



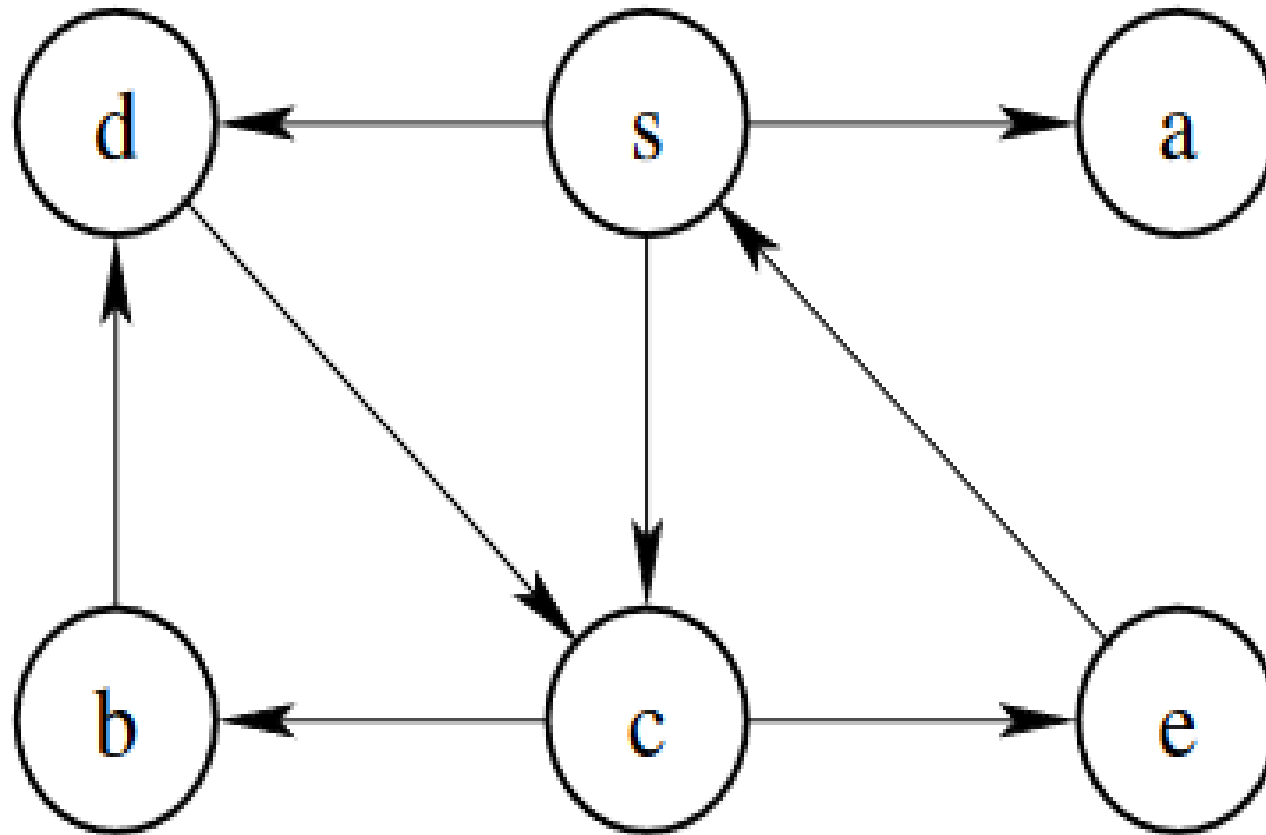
Class Exercise 1

- (a) Draw the Depth-First Search tree of the graph, starting at vertex A, and considering neighbours in increasing alphabetical order.
- (b) Draw the Breadth-First Search tree of the graph, starting at vertex A, and considering neighbours in increasing alphabetical order.



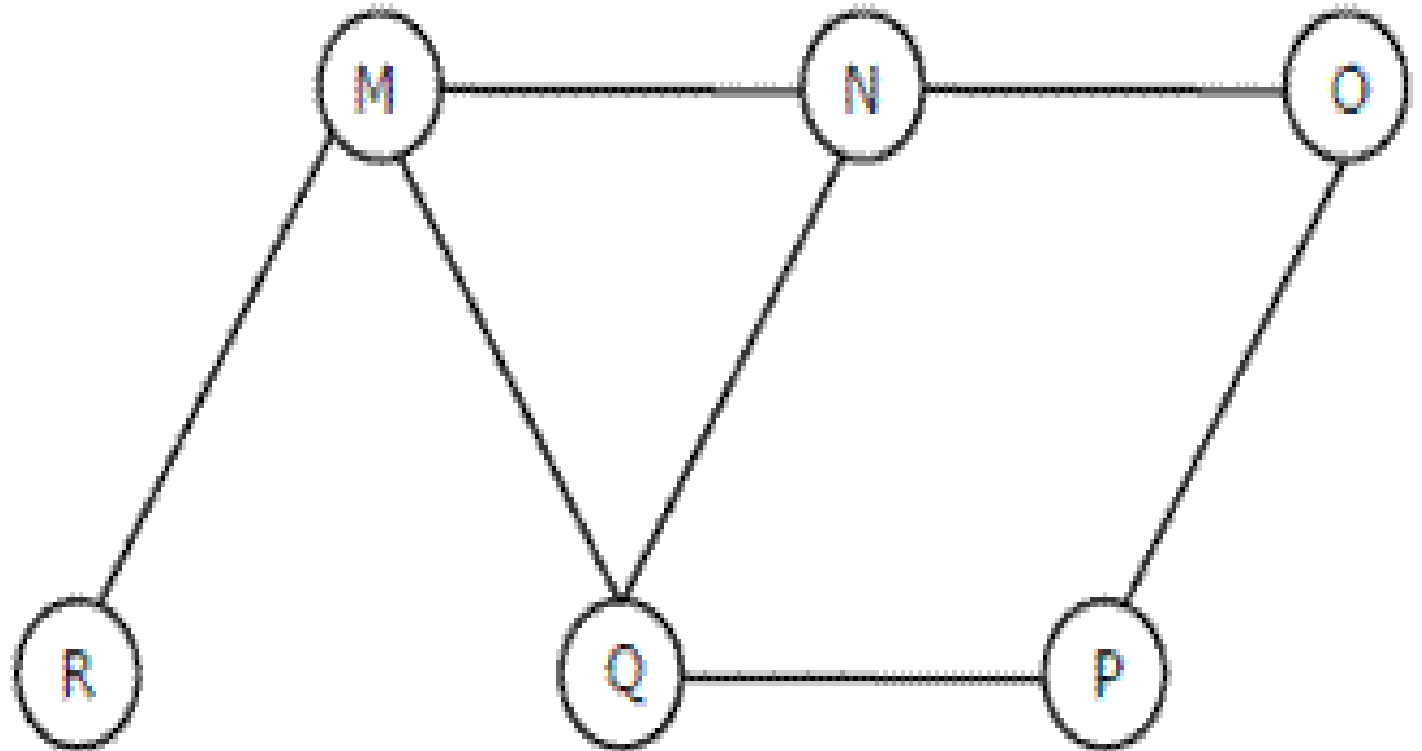
Class Exercise 2

Give the visited node order for each type of graph search (i.e. BFS and DFS), starting with node s for the given graph.



Class Exercise 3

The Breadth First Search algorithm has been implemented using the queue data structure. One possible order of visiting the nodes of the following graph is



- (A) MNOPQR
- (B) NQMPOR
- (C) QMNPOR
- (D) QMNPOR

Class Exercise 4

Consider the following graph Among the following sequences

I) a b h g e f

II) a b g f e h

III) a f h g b e

IV) a f h g e b

Which are DFS

traversals of the above grap

(A) I, II and IV only

(B) I and IV only

(C) II, III and IV only

(D) I, III and IV only

