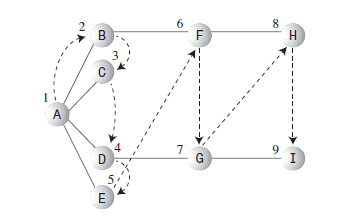
/\*\*\*\*\*\*\*\*\*\* Breadth First Search(BFS) Algorithm \*\*\*\*\*\*\*\*\*\*/



Graph Details :

Vertices :

theGraph.addVertex(‘A’); // 0 (start for dfs)

theGraph.addVertex(‘B’); // 1

theGraph.addVertex(‘C’); // 2

theGraph.addVertex(‘D’); // 3

theGraph.addVertex(‘E’); // 4

theGraph.addVertex(‘F’); // 5

theGraph.addVertex(‘G’); // 6

theGraph.addVertex(‘H’); // 7

theGraph.addVertex(‘I’); // 8

Edges :

(Starting Vertex - A)

theGraph.addEdge(0, 1); // AB

theGraph.addEdge(0, 2); // AC

theGraph.addEdge(0, 3); // AD

theGraph.addEdge(0, 4); // AE

(Starting Vertex - B)

theGraph.addEdge(1, 5); // BF

(Starting Vertex - D)

theGraph.addEdge(3, 6); //DG

(Starting Vertex - F)

theGraph.addEdge(5, 7); //FH

(Starting Vertex - G)

theGraph.addEdge(6, 8); //GI

The Order in which the Vertices are visited are as below sequence :

1st - A

2nd - B

3rd - C

4th - D

5th - E

6th - F

7th - G

8th - H

9th - I

Note : Plot the above Graph before going through the below topic.

Breadth First Search :

The fundamental difference between Depth First Search(DFS) and BFS(Breadth First Search) :

1. DFS is implemented using Stack(Last In First Out Logic). DFS algorithm acts as though it wants to get as far away from the starting point

as quickly as possible.

2. BFS is implemented using Queue(First In First Out Logic). BFS algorithm acts as though it wants to stay as close as possible

to the starting point. It visits all the vertices adjacent to the starting Vertex, and then only goes further afield.

Rules for BFS algorithm to be followed iteratively:

A is the starting Vertex, so you visit it and make it the current Vertex. Then you follow these rules.

**RULE 1:**

Visit the next unvisited vertex(if there is one) that is adjacent to the current vertex., mark it and insert it into the queue.

Note - In a Queue, the elements are always inserted at the rear end and removed at the front end. It follows First In First Out Logic.

**RULE 2:**

If you can not carry out RULE 1 because there are no more unvisited vertices adjacent to the current vertex, remove a vertex from the Queue if possible and make it the current vertex.

Note - Removal of items from a Queue is always done at the Front end.

**RULE 3:**

If you can not carry out RULE 2 because the Queue is empty, you are done.