BREADTH FIRST SEARCH ALGORITHM

1 Creates a tree by traversing all immediate neigbour vertices before reaching out to others.

2 Outputs a shallow tree.

Pre-Conditions:

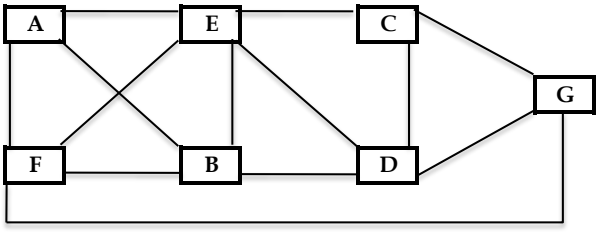
1 Undirected or directed graph

Required Data Structures:

1 A Queue [FIFO Structure]

2 A 1-D array, Path[] to store the vertices in the BFS traversal.

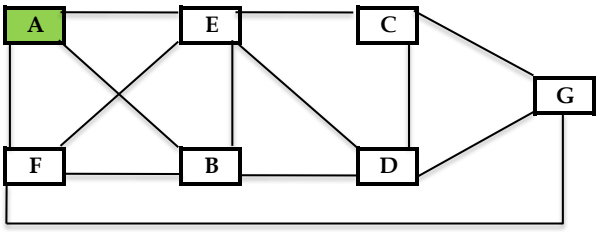
3 A 1-D array Visited[] to keep track of already visited vertices.



	A	B	C	D	E	F	G
A	0	1	0	0	1	1	0
B	1	0	0	1	1	1	0
C	0	0	0	1	1	0	1
D	0	1	1	0	1	0	1
E	1	1	1	1	0	1	0
F	1	1	0	0	1	0	1
G	0	0	1	1	0	1	0

Adjacency Matrix

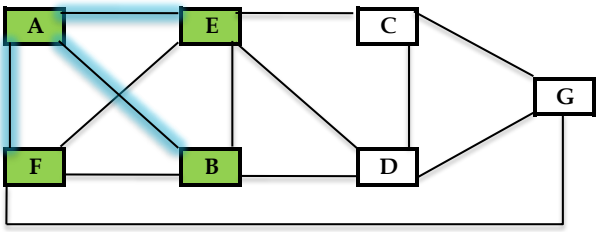
Initial State



SRC := **A**

VISITED[]	A	B	C	D	E	F	G
	T	F	F	F	F	F	F

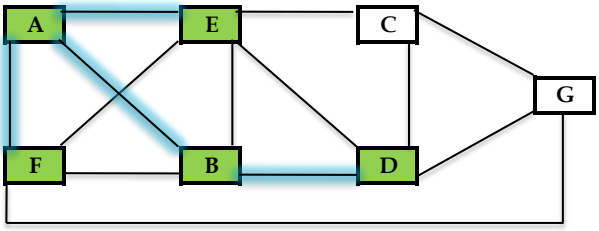
PATH[]						
A	-	-	-	-	-	-



Dequeue **A** Enqueue: B, E, F

	A	B	C	D	E	F	G
A	0	1	0	0	1	1	0
VISITED[]	T	T	F	F	T	T	F

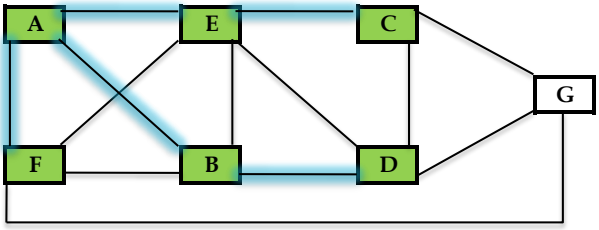
PATH[]						
A	B	E	F	-	-	-



Dequeue **B** Enqueue: D

	A	B	C	D	E	F	G
B	1	0	0	1	1	1	0
VISITED[]	T	T	F	T	T	T	F

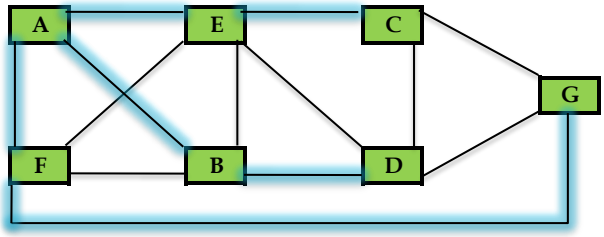
PATH[]						
A	B	E	F	D	-	-



Dequeue **E** Enqueue: C

	A	B	C	D	E	F	G
E	1	1	1	1	0	1	0
VISITED[]	T	T	T	T	T	T	F

PATH[]	A	B	E	F	D	C	-



Dequeue **F** Enqueue: G

	A	B	C	D	E	F	G
F	1	1	0	0	1	0	1
VISITED[]	T	T	T	T	T	T	T

PATH[]	A	B	E	F	D	C	G

By : **Prof. D. A. Borikar**
CSE, RCOEM, NAGPUR
AY - 2019-20