DEPTH FIRST SEARCH ALGORITHM

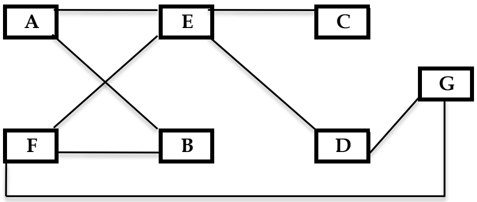
1 Creates a tree by traversing all immediate neigbour vertices before reaching out to others.
2 Outputs a deep tree.

Pre-Conditions:

1 Undirected or directed graph

Required Data Structures:

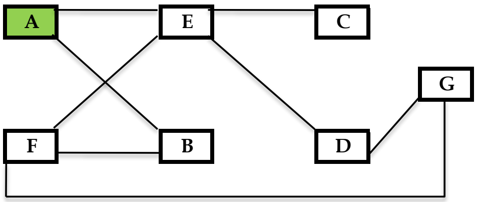
1 A Stack [LIFO Structure]
2 A 1-D array, Path[] to store the vertices in the DFS traversal.
3 A 1-D array Visited[] to keep track of already visited vertices.



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

Adjacency Matrix

Initial State Call **DFS** (G[][| V |], **A**)



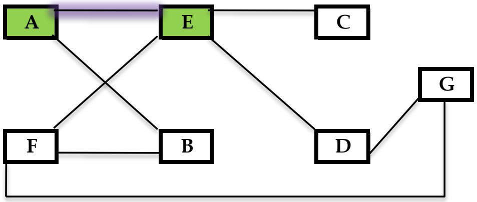
Call Stack
DFS(G, **A**)

SRC := A

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

V_Chosen : **E** [Call DFS with SRC := E]

Phase-1 Call **DFS** (G[][| V |], **E**)



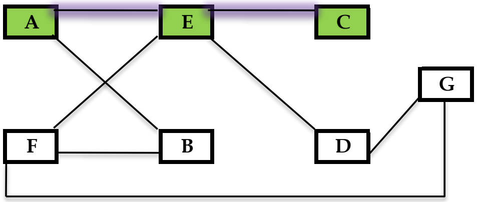
DFS(G, **E**)
DFS(G, **A**)

SRC := E

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

V_Chosen : **C** [Call DFS with SRC := C]

Phase-2 Call **DFS** (G[][| V |], **C**)



DFS(G, **C**)
DFS(G, **E**)
DFS(G, **A**)

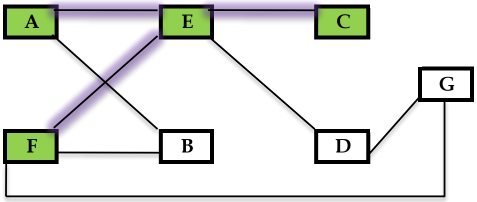
SRC := C

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

DFS(C) Terminates, returns to execute DFS(E)

V_Chosen : **F** [Call DFS with SRC := F]

Phase-3 Call **DFS** (G[][| V |], **F**)



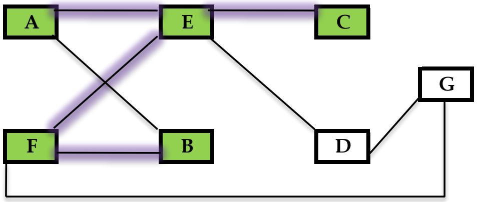
DFS(G, **F**)
DFS(G, **E**)
DFS(G, **A**)

SRC := F

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

V_Chosen : **B** [Call DFS with SRC := B]

Phase-4 Call **DFS** (G[][|V|], **B**)



DFS(G, **B**)
DFS(G, **F**)
DFS(G, **E**)
DFS(G, **A**)

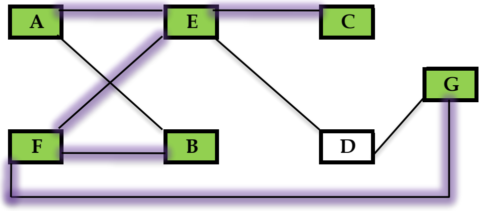
SRC := B

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

DFS(B) Terminates, returns to execute DFS(F)

V_Chosen : **D** [Call DFS with SRC := D]

Phase-5 Call **DFS** (G[][|V|], **G**)



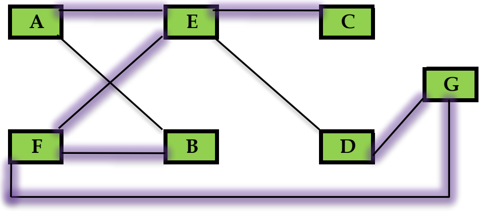
DFS(G, **G**)
DFS(G, **F**)
DFS(G, **E**)
DFS(G, **A**)

SRC := B

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

V_Chosen : **D** [Call DFS with SRC := D]

Phase-6 Call **DFS** (G[][|V|], **D**)



DFS(G, **D**)
DFS(G, **G**)
DFS(G, **F**)
DFS(G, **E**)
DFS(G, **A**)

SRC := D

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

As All Vertices are Traversed…

DFS(D) Terminates, returns to execute DFS(G)

DFS(G) Terminates, returns to execute DFS(F)

DFS(F) Terminates, returns to execute DFS(E)

DFS(E) Terminates, returns to execute DFS(A)

DFS(A) Completes, the Search is Printed.

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