

Lecture 12

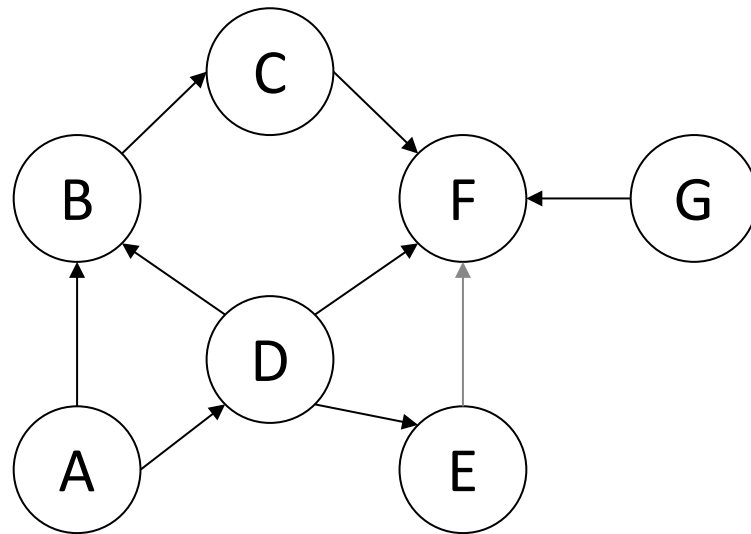
Graphs

Exercises ANS

Department of Computer Science
Hofstra University

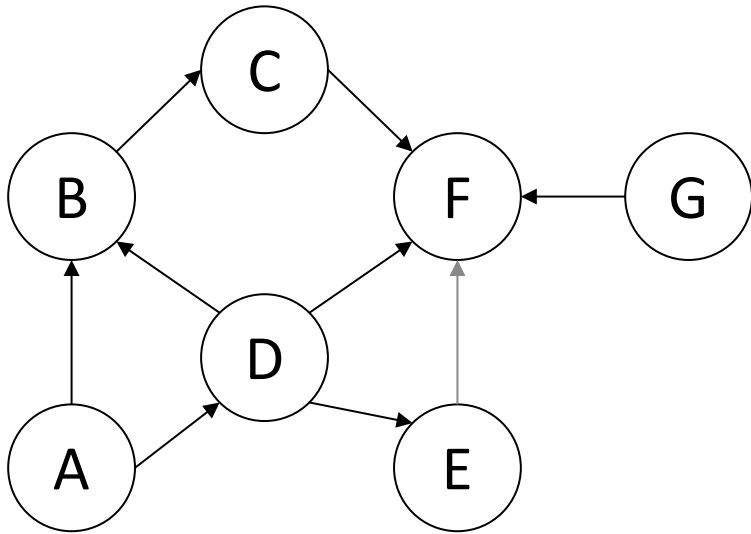
Q1. Adjacency matrix and adjacency list

- Write out the adjacency matrix and adjacency list for the directed graph.

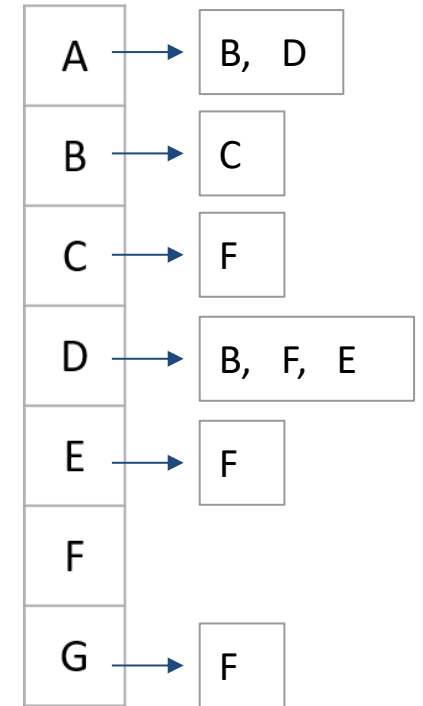


Q1. Adjacency matrix and adjacency list ANS

- Write out the adjacency matrix and adjacency list for the directed graph.

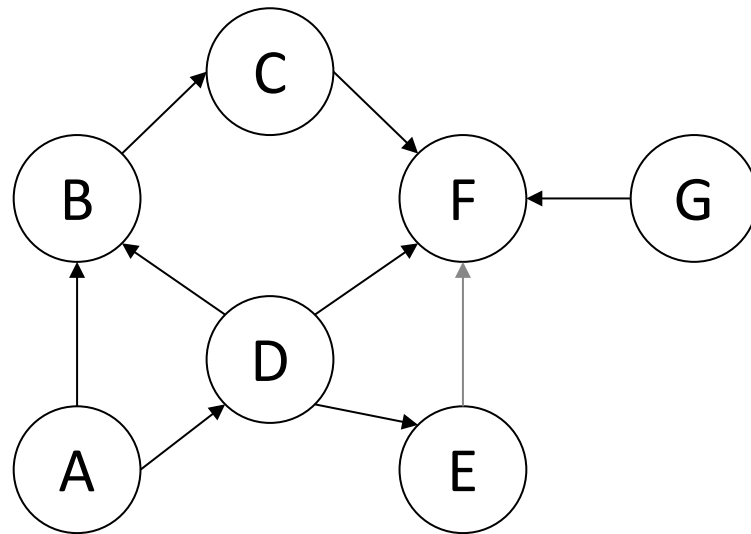


	A	B	C	D	E	F	G
A	0	1	0	1	0	0	0
B	0	0	1	0	0	0	0
C	0	0	0	0	0	1	0
D	0	1	0	0	1	1	0
E	0	0	0	0	0	1	0
F	0	0	0	0	0	0	0
G	0	0	0	0	0	1	0



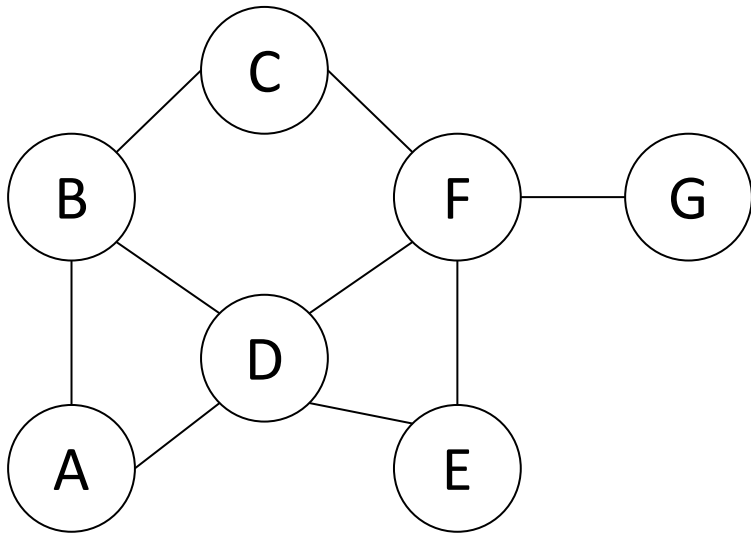
Q2. Adjacency matrix and adjacency list

- Write out the adjacency matrix and adjacency list for the undirected graph.

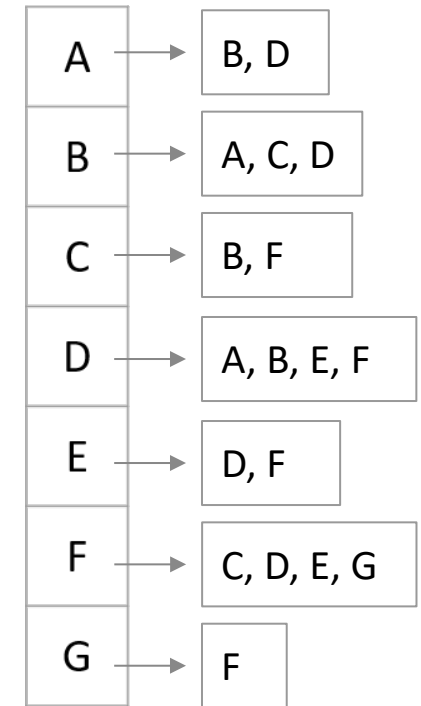


Q2. Adjacency matrix and adjacency list ANS

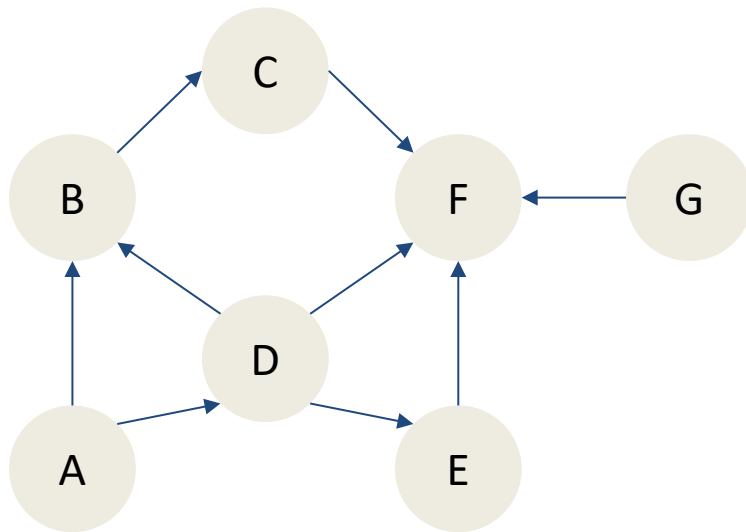
- Write out the adjacency matrix and adjacency list for the directed graph.



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



Q3: Pre-Order & Post-Order Traversals

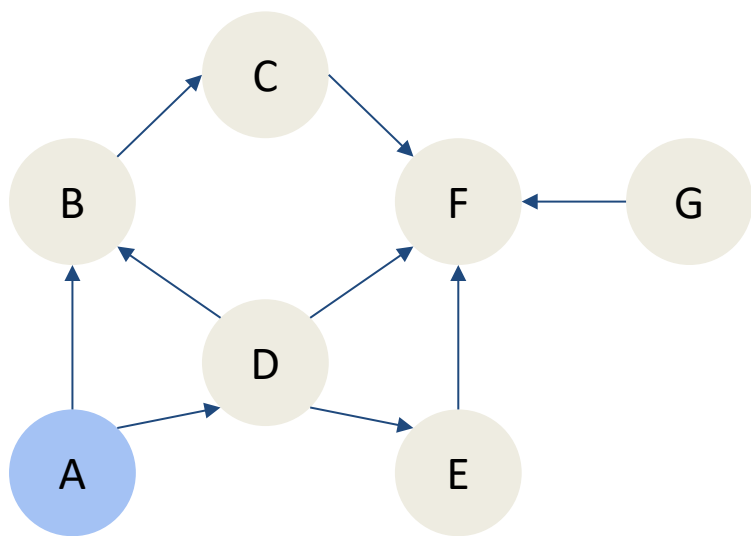


DFS Pre-Order:

DFS Post-Order:

Stack:

We use a stack-based implementation instead of recursive function calls

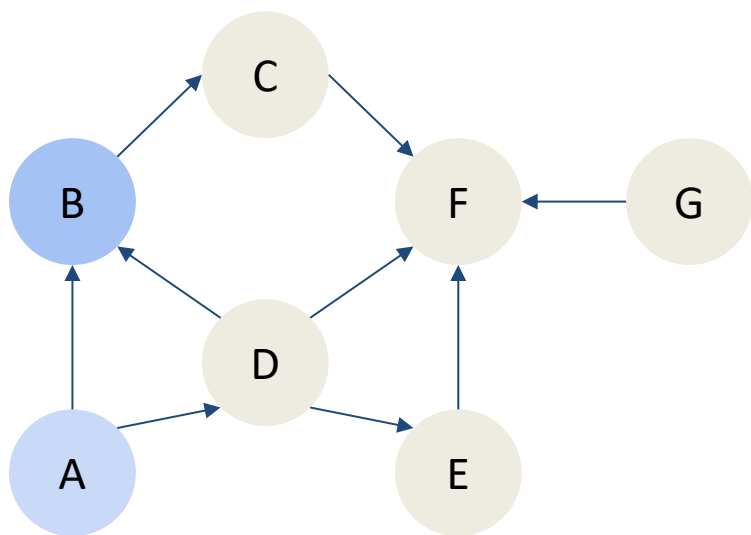


Stack: A

DFS Pre-Order:

A

DFS Post-Order:

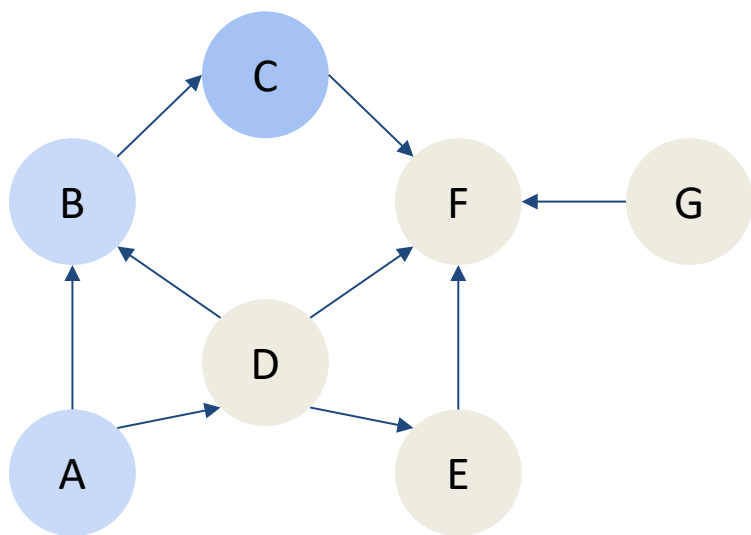


Stack: A, B

DFS Pre-Order:

A, B

DFS Post-Order:

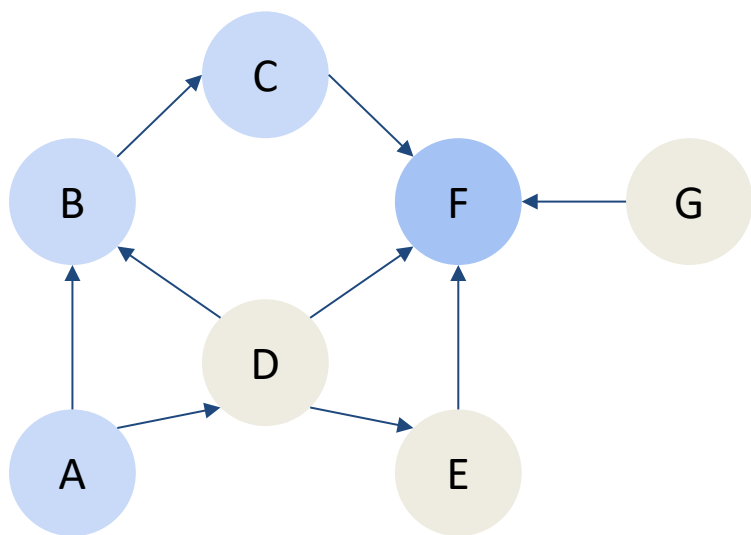


Stack: A, B, C

DFS Pre-Order:

A, B, C

DFS Post-Order:

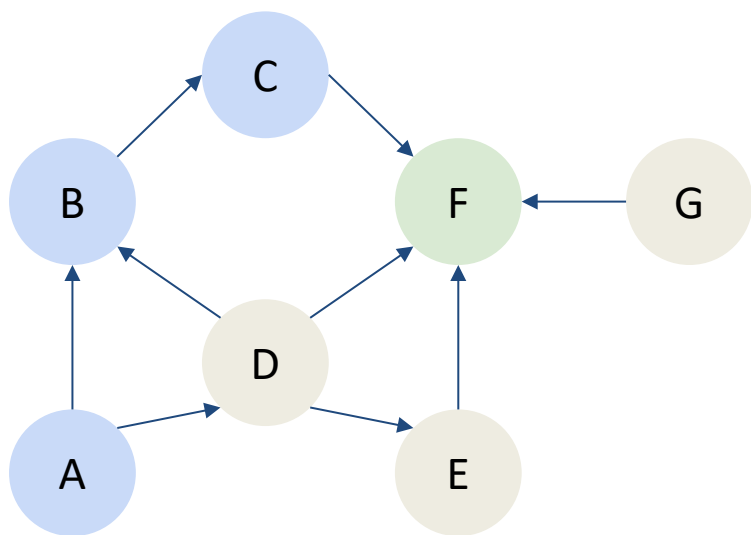


Stack: A, B, C, F

DFS Pre-Order:

A, B, C, F

DFS Post-Order:



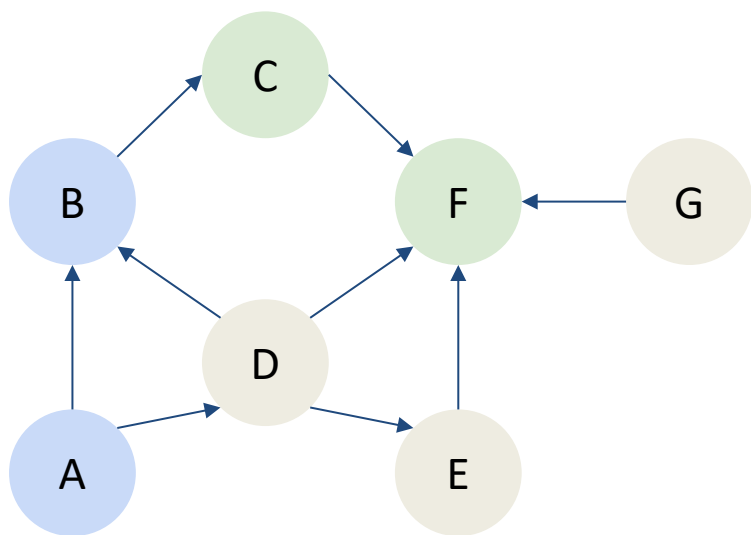
Stack: A, B, C

DFS Pre-Order:

A, B, C, F

DFS Post-Order:

F



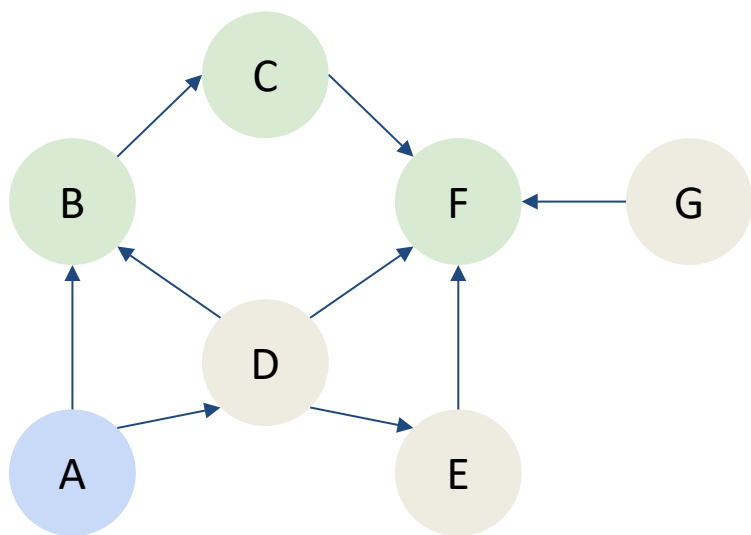
Stack: A, B

DFS Pre-Order:

A, B, C, F

DFS Post-Order:

F, C



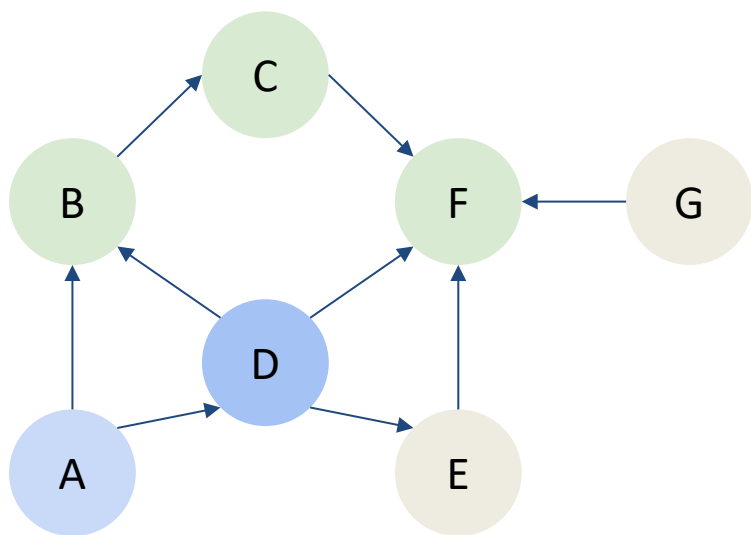
Stack: A

DFS Pre-Order:

A, B, C, F

DFS Post-Order:

F, C, B



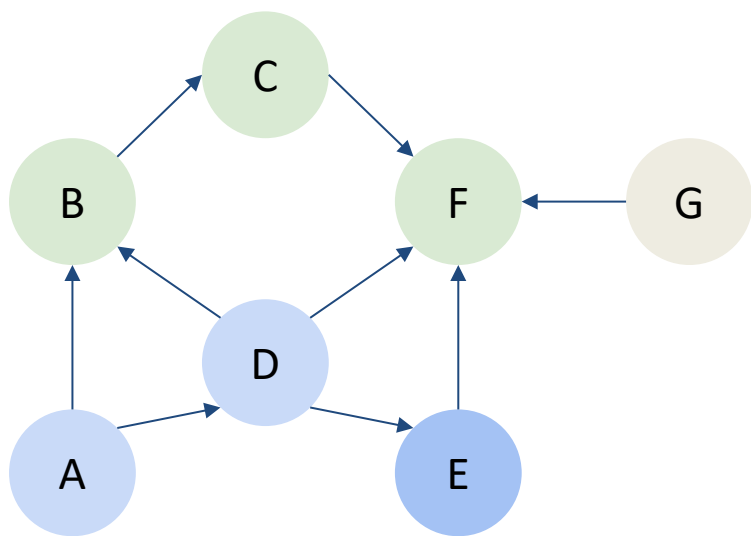
Stack: A, D

DFS Pre-Order:

A, B, C, F, D

DFS Post-Order:

F, C, B



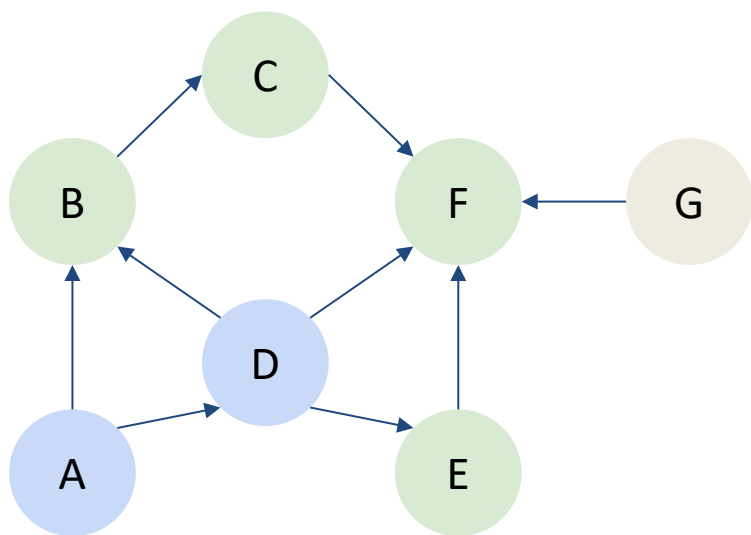
Stack: A, D, E

DFS Pre-Order:

A, B, C, F, D, E

DFS Post-Order:

F, C, B,



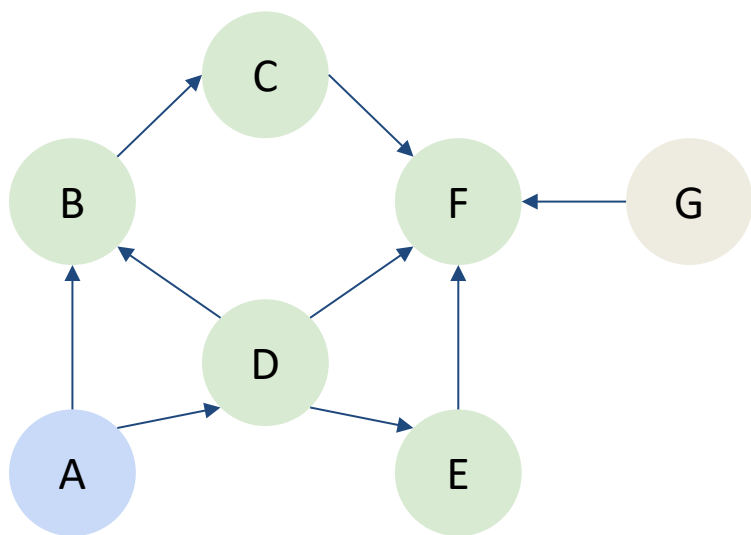
Stack: A, D

DFS Pre-Order:

A, B, C, F, D, E

DFS Post-Order:

F, C, B, E



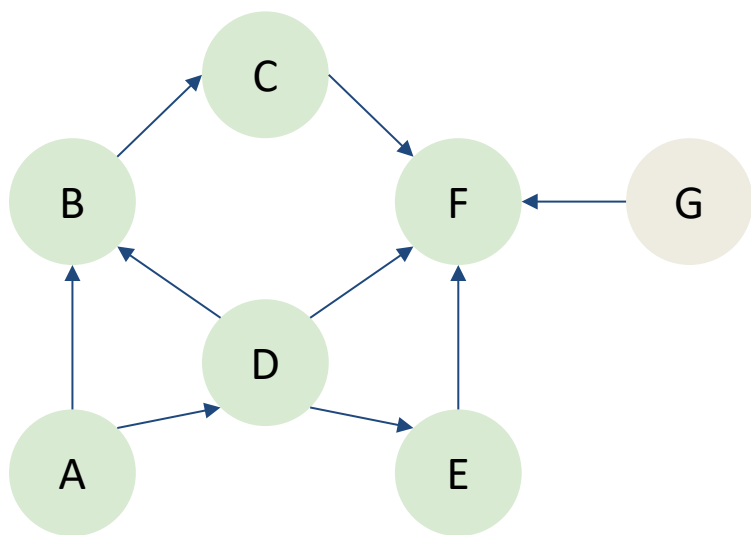
Stack: A,

DFS Pre-Order:

A, B, C, F, D, E

DFS Post-Order:

F, C, B, E, D



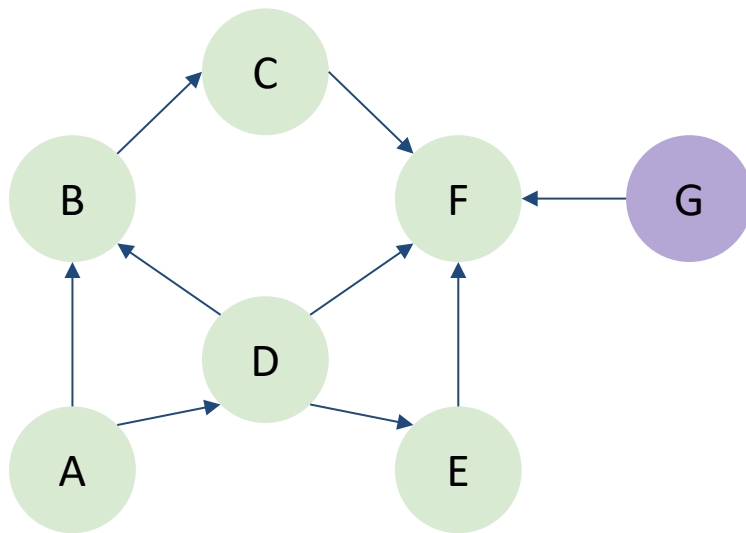
Stack:

DFS Pre-Order:

A, B, C, F, D, E

DFS Post-Order:

F, C, B, E, D, A



Stack:

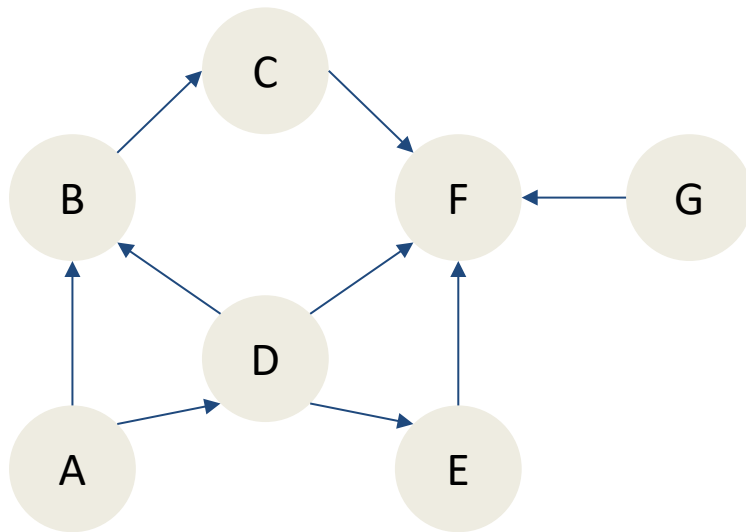
DFS Pre-Order:
A, B, C, F, D, E, G

DFS Post-Order:
F, C, B, E, D, A, G

Topological Sort (reverse of DFS
Post-Order):
G, A, D, E, B, C, F

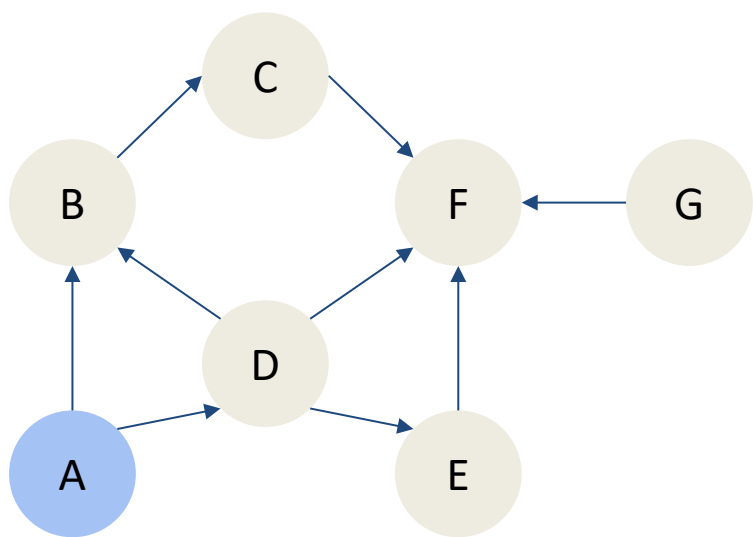
* if we allow DFS to restart on unmarked nodes, G would be added to the stack (and forming the last element in both the preorder and postorder traversals)

Q4. BFS



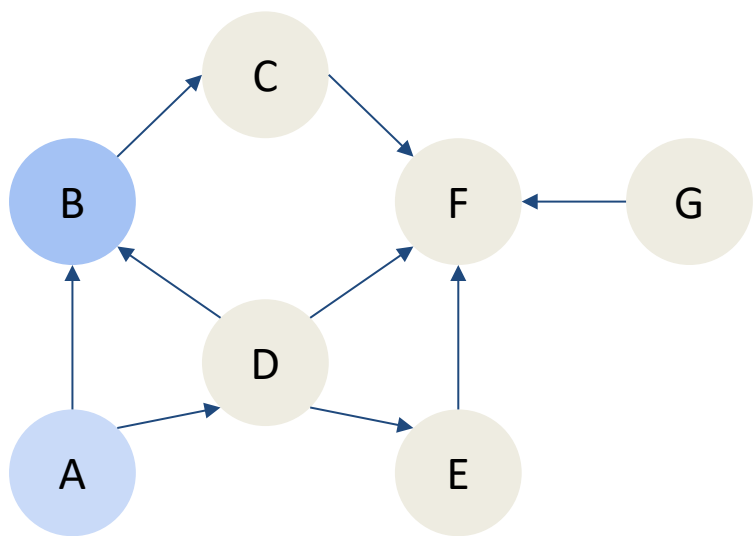
BFS:

Queue: A



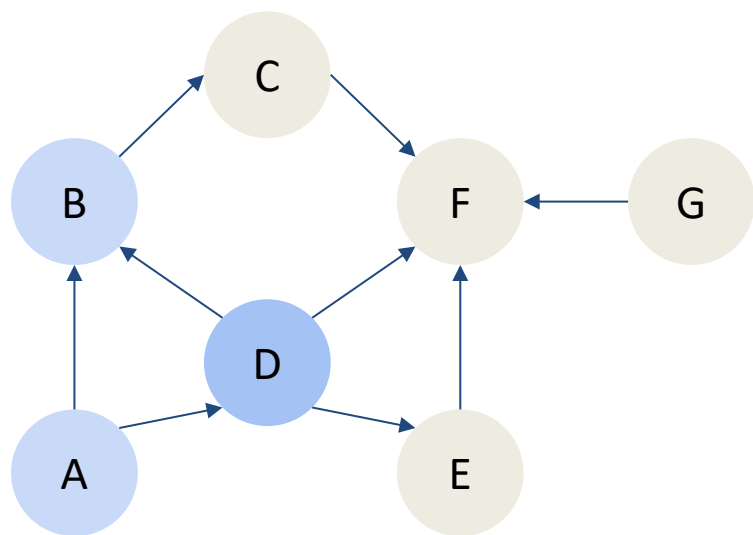
BFS:
A

Queue: B D



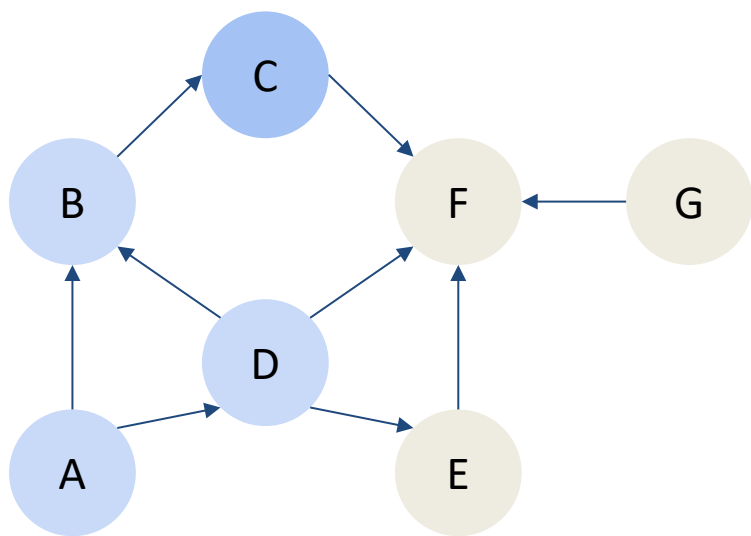
BFS:
A B

Queue: D C



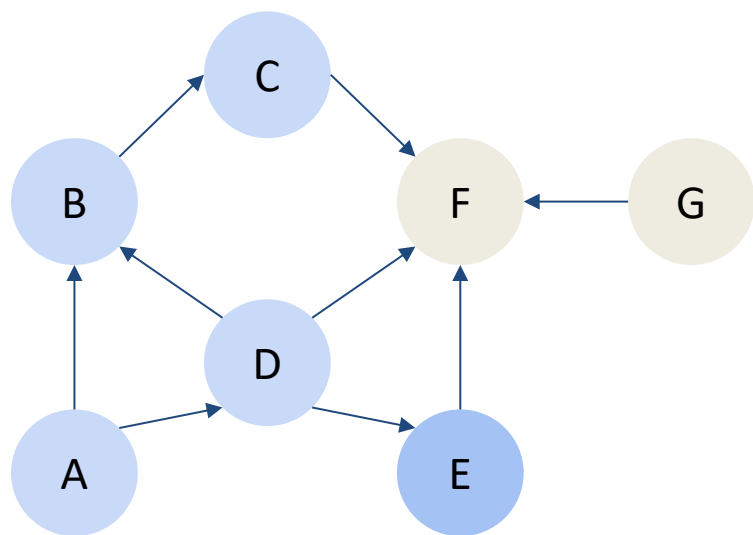
BFS:
A B D

Queue: C E F



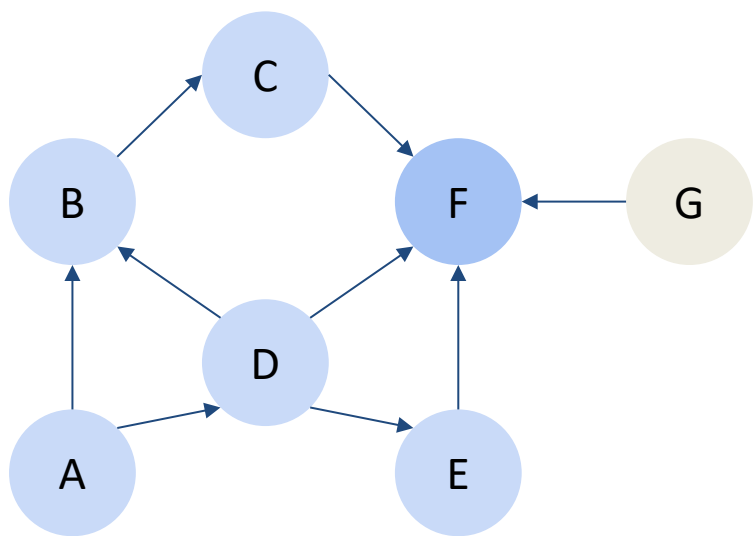
BFS:
A B D C

Queue: E F



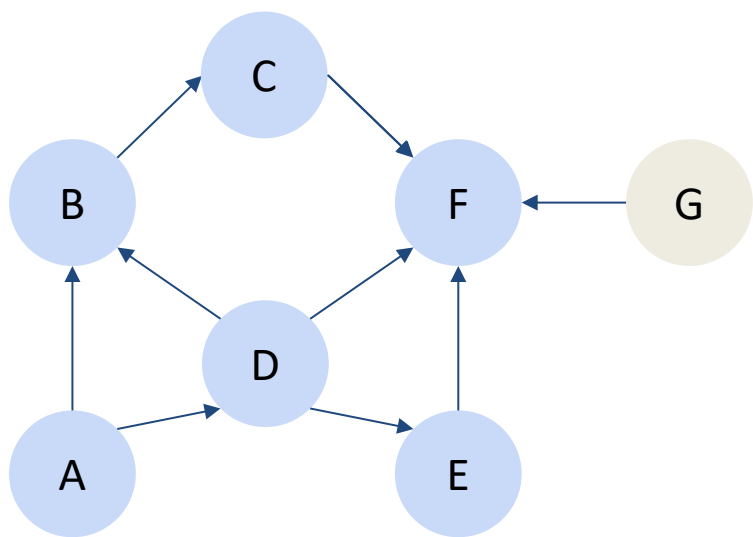
Queue: F

BFS:
A B D C E



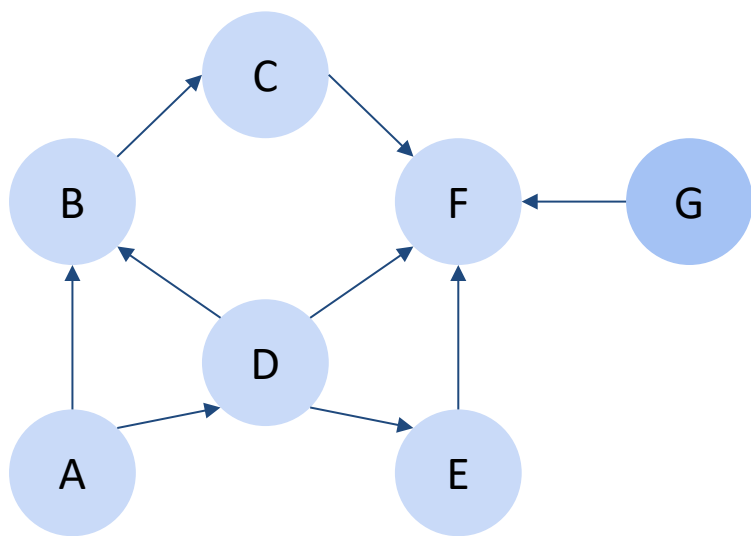
BFS:
A B D C E F

Queue:



Queue: G

BFS:
A B D C E F

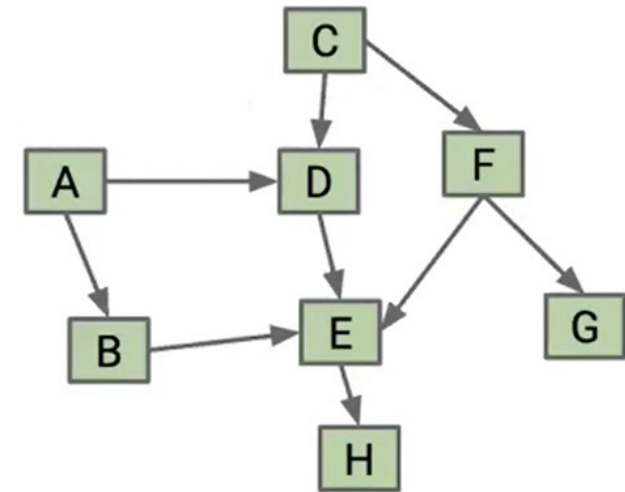


BFS:
A B D C E F G

Queue:

Q5. Graph Traversals

- Starting from node A:
 - Pre-order traversal:
 - Post-order traversal:
 - Topological Sort:
 - BFS:
- Starting from node C:
 - Pre-order traversal:
 - Post-order traversal:
 - Topological Sort:
 - BFS:



Q5. Graph Traversals ANS

- Starting from node A:
 - Pre-order traversal: (A, B, E, H, D, C, F, G)
 - Post-order traversal: (H, E, B, D, A, G, F, C)
 - Topological Sort: (C, F, G, A, D, B, E, H)
 - BFS: (A, B, D, E, H, C, F, G)
- Starting from node C, '
 - Pre-order traversal: (A, B, E, H, D, C, F, G)
 - Post-order traversal: (H, E, B, D, A, G, F, C)
 - Topological Sort: (C, F, G, A, D, B, E, H)
 - BFS: (C, D, F, E, G, H, A, B)

