CS & IT ENGINEERING

Algorithms

Graph Algorithms

Lecture No.- 02

By- Aditya Jain sir

Recap of Previous Lecture

Topic

Topic

DFS on Undirected graph

=

BFS

Topics to be Covered

- Topic
- Topic
- Topic

DFS on Directed Graph

About Aditya Jain sir

1. Appeared for GATE during BTech and secured AIR 60 in GATE in very first attempt - City topper
2. Represented college as the first Google DSC Ambassador.
3. The only student from the batch to secure an internship at Amazon. (9+ CGPA)
4. Had offer from IIT Bombay and IISc Bangalore to join the Masters program
5. Joined IIT Bombay for my 2 year Masters program, specialization in Data Science
6. Published multiple research papers in well known conferences along with the team
7. Received the prestigious excellence in Research award from IIT Bombay for my Masters thesis
8. Completed my Masters with an overall GPA of 9.36/10
9. Joined Dream11 as a Data Scientist
10. Have mentored 12,000+ students & working professions in field of Data Science and Analytics
11. Have been mentoring & teaching GATE aspirants to secure a great rank in limited time
12. Have got around 27.5K followers on Linkedin where I share my insights and guide students and professionals.

Telegram

Telegram Link for Aditya Jain sir: https://t.me/AdityaSir_PW

Topic : Graph Algorithms

V. imP

DFS in directed graphs:

Terminologies:-

Whenever we perform DFS on a directed graph, it can lead to the different types of edges listed before.

1. Tree Edge:

Tree edge are those edges that are present in the DFS spanning tree tree/forest:

The edges that are not present in DFS spanning tree/forest can be 3 types:

1. Forward
2. Backward
3. Cross

Topic : Graph Algorithms

2. Forward edge:-

   It is the edge that is going form a node to its non-child descendent.

3. Back/Backward edge:

   This is an edge from a node to its ancestor

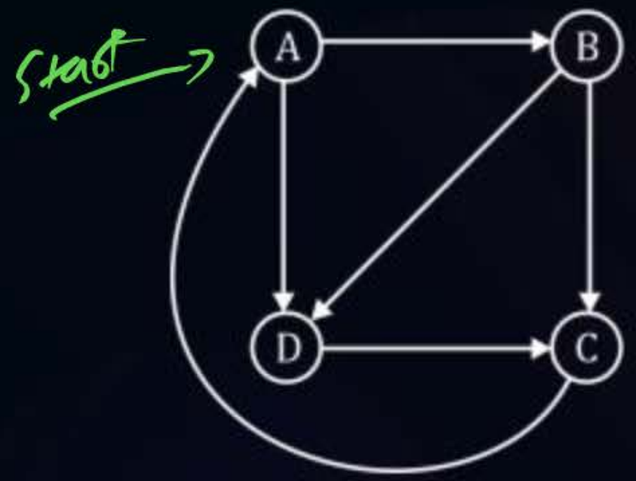
4. Cross edge:-

   This is the edge from one node to some other node that is neither its ancestor nor its descendent

Topic : Graph Algorithms

E.g.1. Given G(V,E)

Apply DGS starting with A

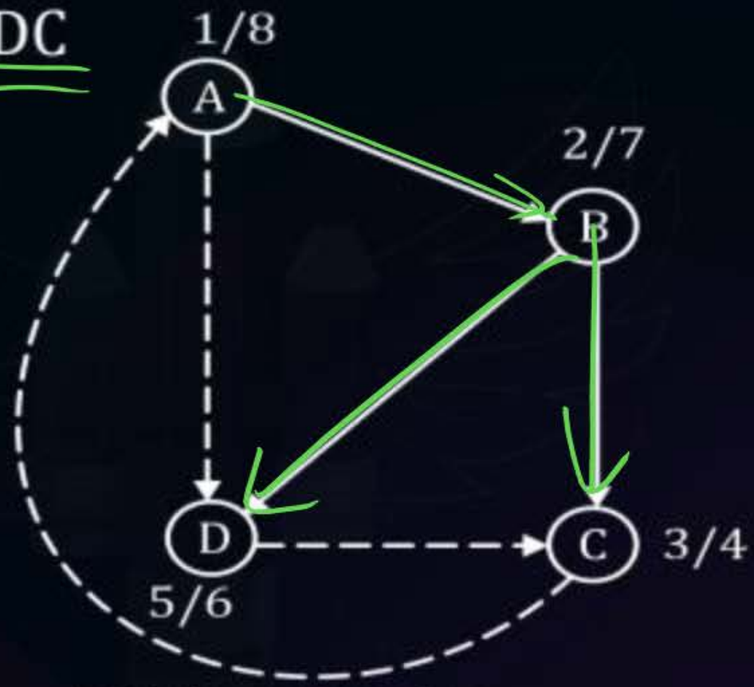


1. DFS Sorting

Types of edges

1. Tree edges:-AB, BC, BD
2. Forward edge:-AD
3. Backward edge:-CA
4. Cross edge:-DC

DFS ST →



DFS Spanning Tree

Topic : Graph Algorithms

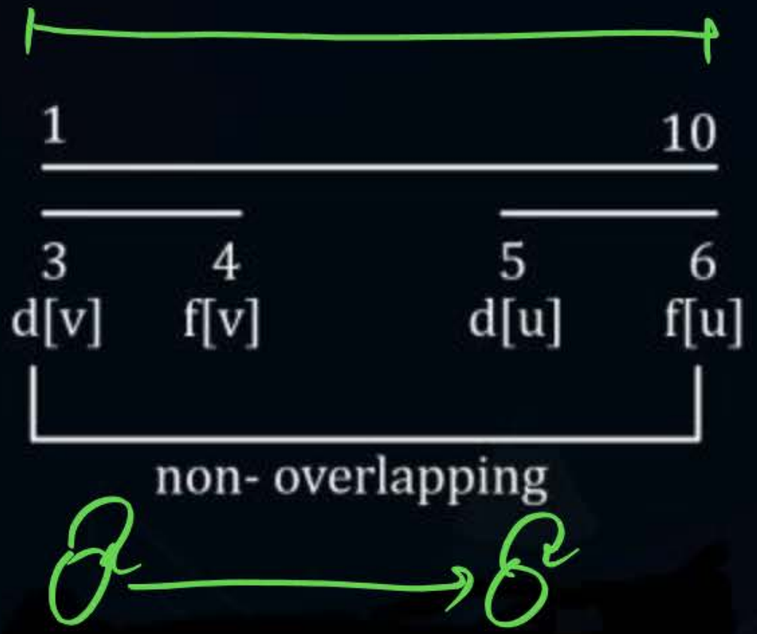
Property 1.

Cross edge

D→ C

D: u : 5/6

C: V : 3/4



Topic : Graph Algorithms

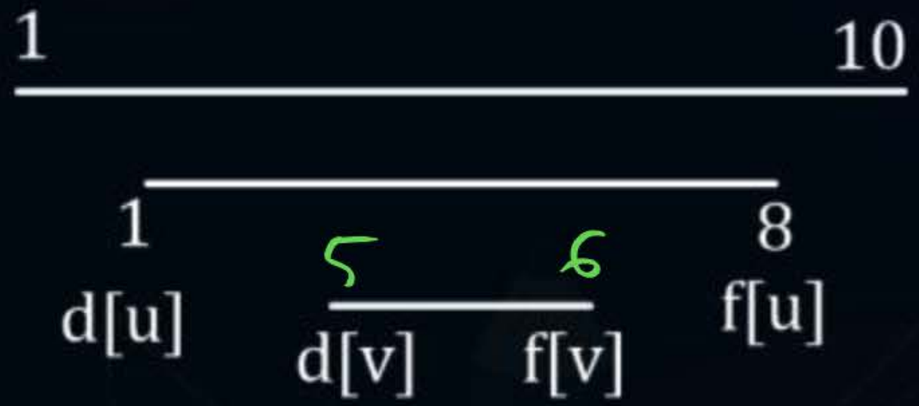
E.g.3.

Forward edge A → D

u: A: d[u]/ f[u]: 1/8

v: D: d[v]/ f[v]: 5/6

u → v : forward edge.



Topic : Graph Algorithms

Property - 2.

Backward edge: C → A

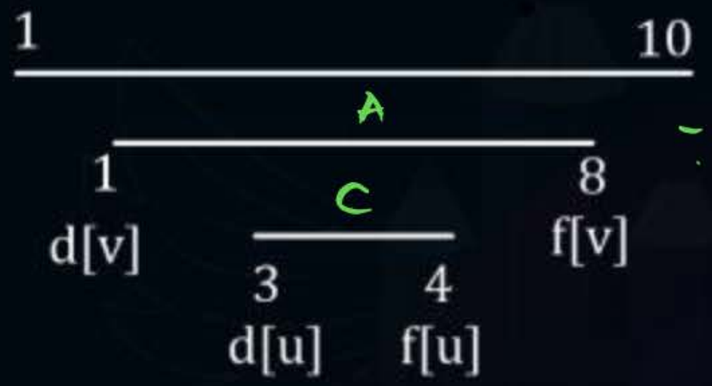
C : u → 3/4

A : v → 1/8

C: d[u] /f[u]: 3/4

A: d[v] /f[v]: 1/8

u → v back edge



u → v

Topic : Graph Algorithms

Note. For same graph for same starting vertex different DFS spanning Trees are possible and hence, the types of edges can also be different.

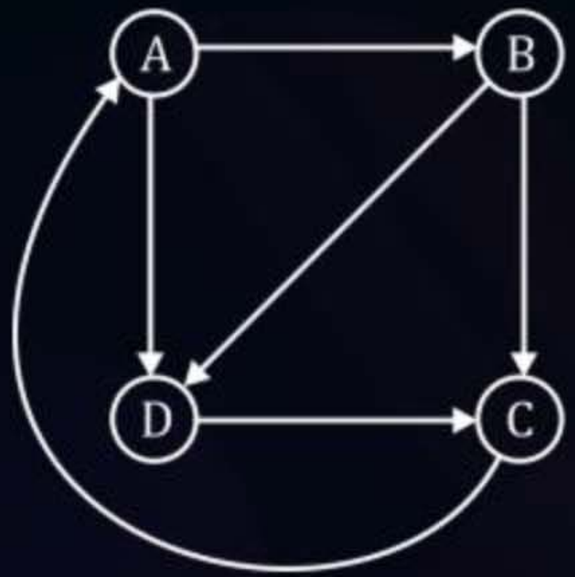
Obs: Tree edge
+
Forward edge
+
Backward edges
+
Cross edge

} Graph

Topic : Graph Algorithms

E.g. 4.

Given G (V,E)

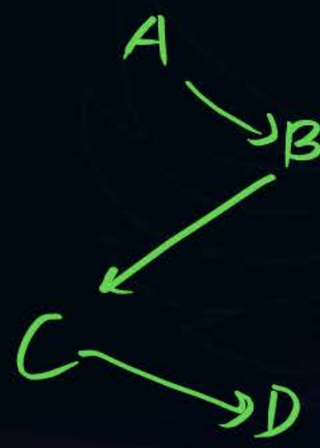


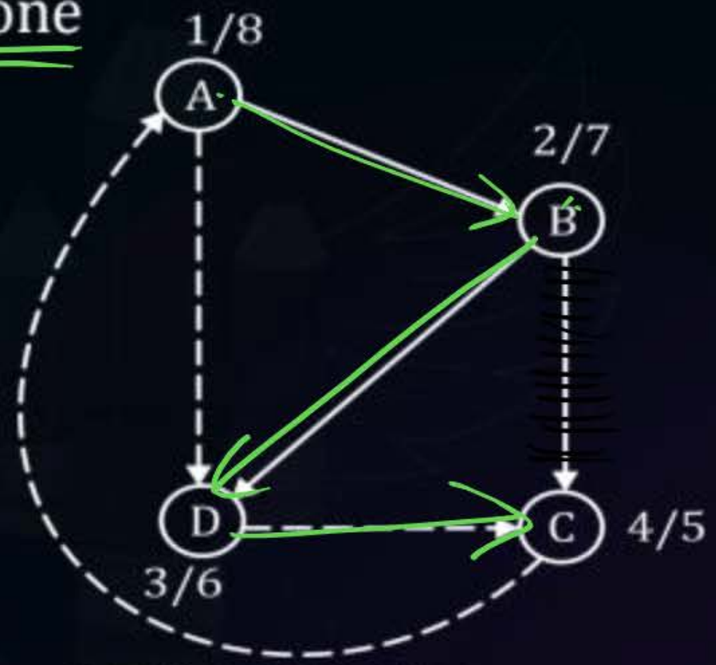
Apply DFS starting at A.

2. DFS Sorting at A

Types of edges

1. Tree edges:-AB, BD, DC
2. Forward edge:-AD, BC
3. Backward edge:-CA
4. Cross edge:-None





DFS Spanning Tree

Topic : Graph Algorithms

Imp Properties

In any Depth-First Search of a (directed or undirected) graph G = (V, E), for any two vertices u and v exactly one of the following three conditions holds:

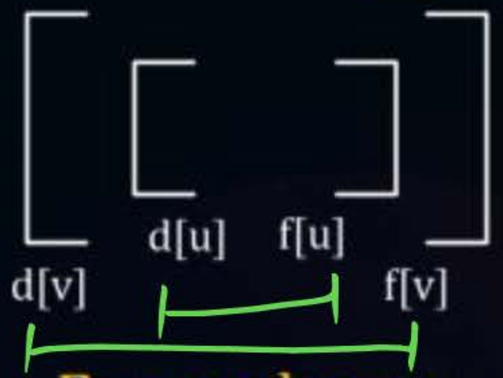
I. The intervals [d[u]], f[u]] and d[v], f[v] are entirely disjoint, and neither u nor v is a descendent of the other in the depth-first forest,

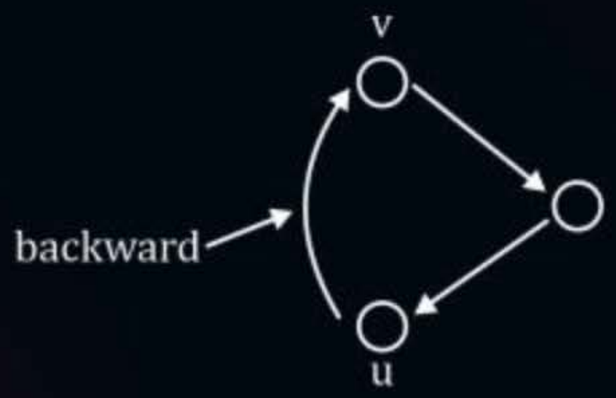
II. The interval [d[u], f[u]] is contained entirely within the interval [d[v], f[v]], and u is a descendent of v in a depth-first tree, or

III. The interval [d[v], f[v]] is contained entirely within the interval [d[u], f[u]], and v is a descendent of u in a depth-first tree.

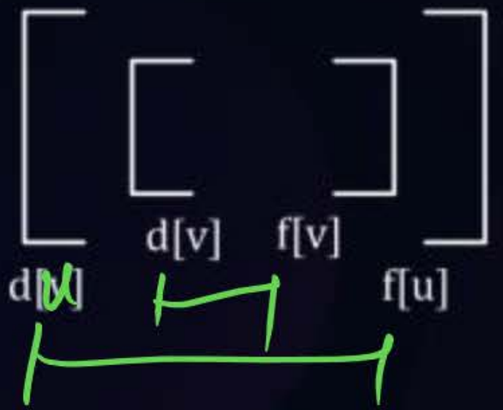
Topic : Graph Algorithms

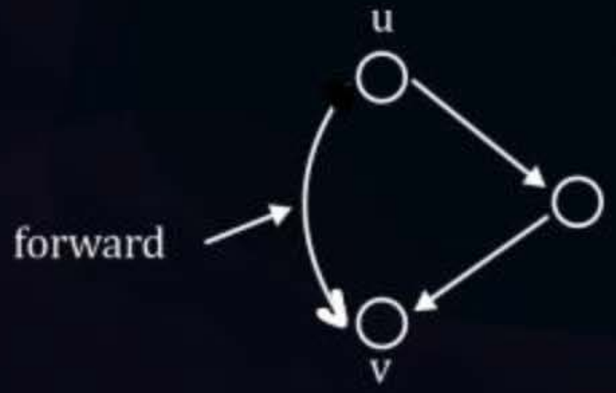
2. Backward u→ v:

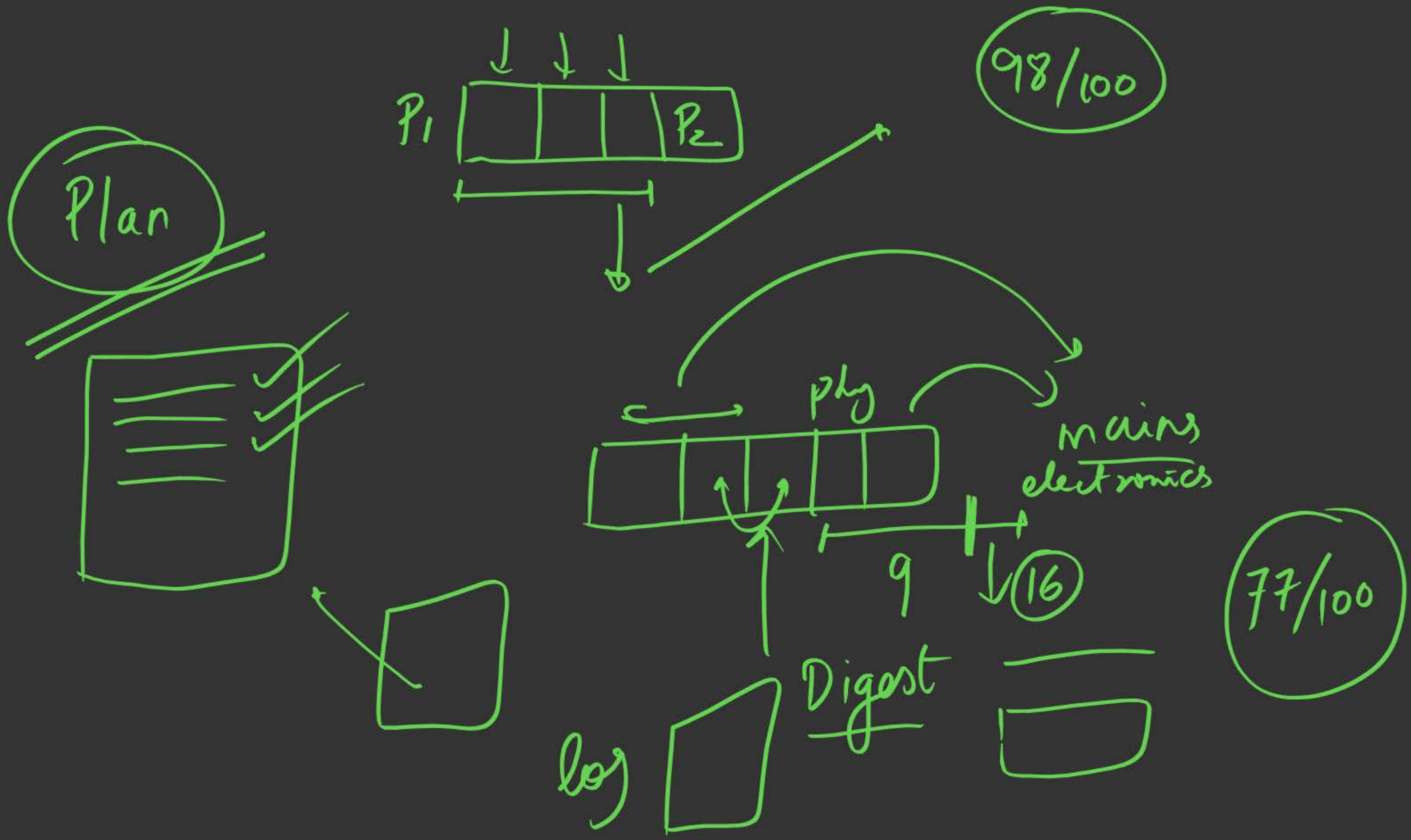




3. Forward u→ v



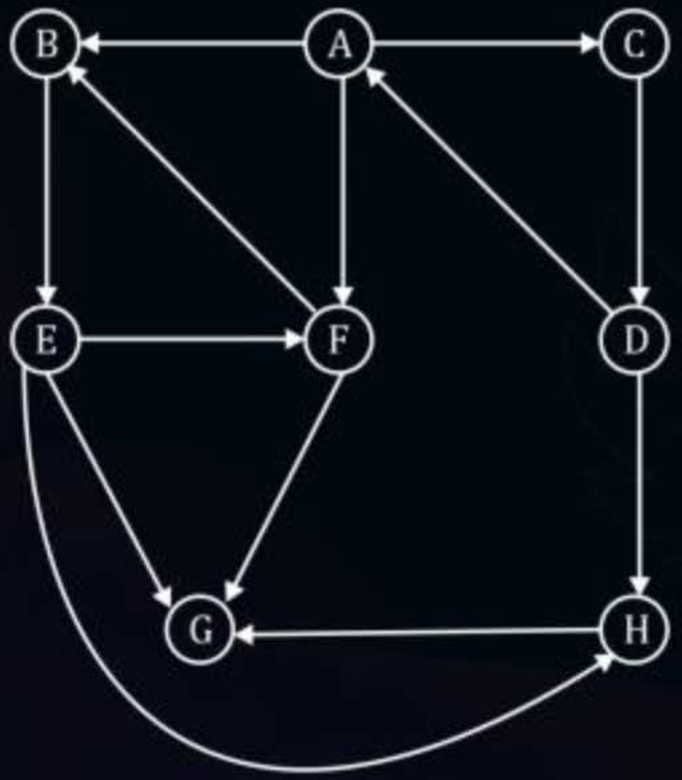



P1
P2
98/100
Plan
ply
mains
electronics
9
16
77/100
Digest
log

Topic : Graph Algorithms

Test: End to end practice:

Give: G (V,E)



Apply DFS
start at A.

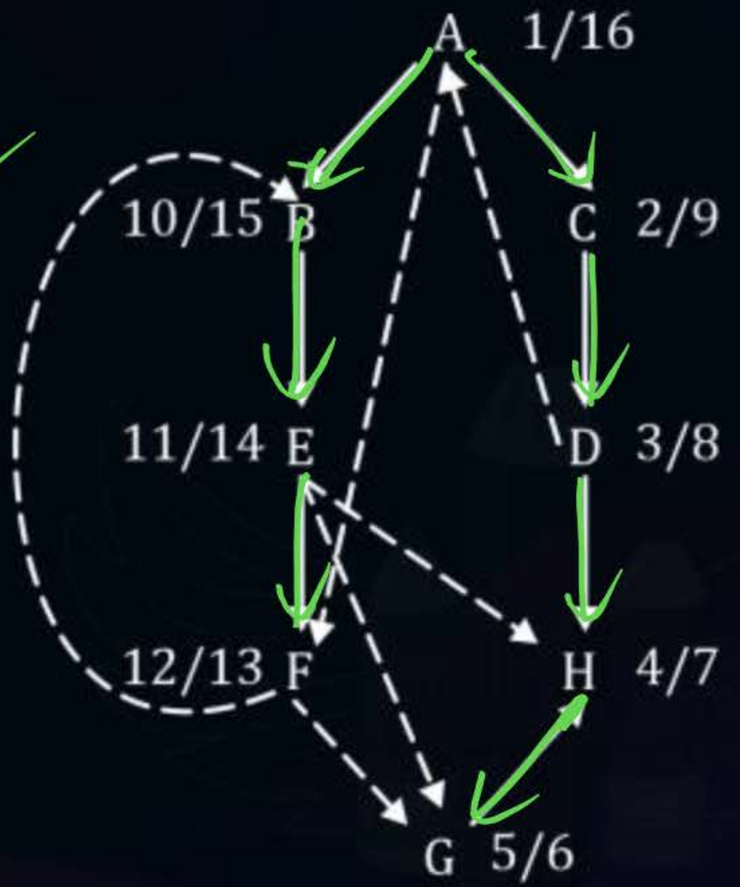
e=13

Topic : Graph Algorithms

DFS starting at A

Type of edges:

1. Tree edges: AC, CD, DH,HG AB,BE, EF.
2. Froward edges: AF
3. Backward edges: DA, FB
4. Cross edge: FG, EG, EH



8 x 2 = 16

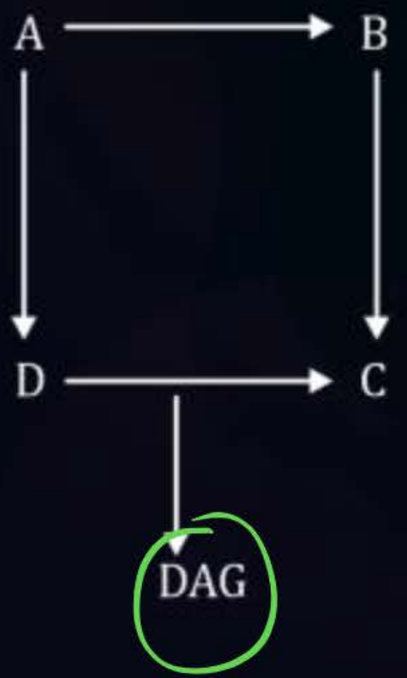
13

7+1+2+3

Topic : Graph Algorithms

DFS in directed a cyclic graph: (DAG)

DAG: Directed graph without any cycle in it.



Topic : Graph Algorithms

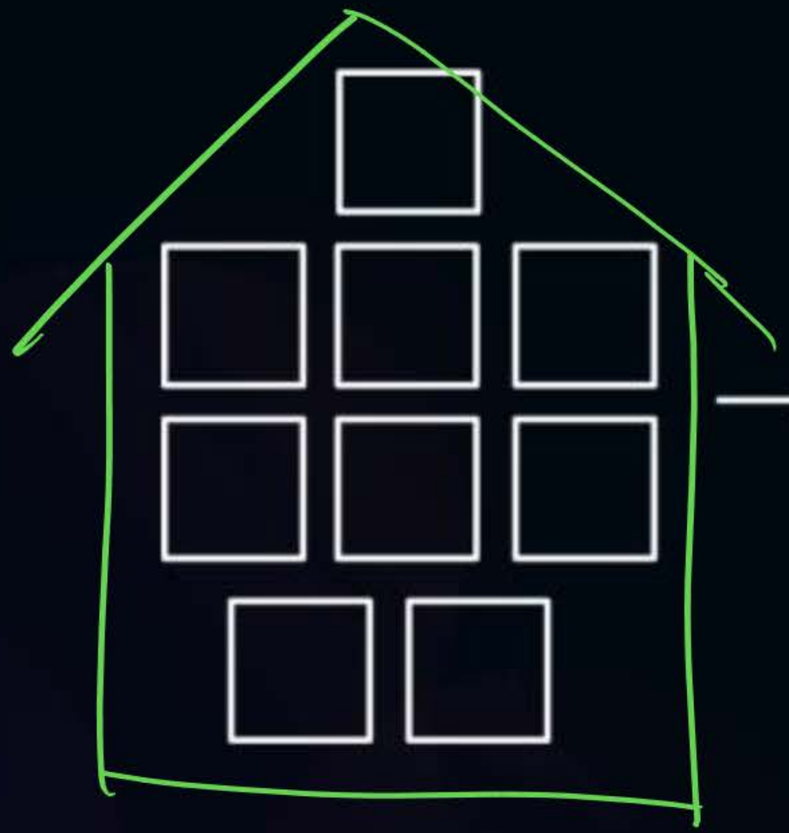
Topological Sort: Application of DFS on a directed a cycle graph DAG

Definition:

It is the linear order of vertices/nodes that represent some activities that has some precedence.

Topic : Graph Algorithms

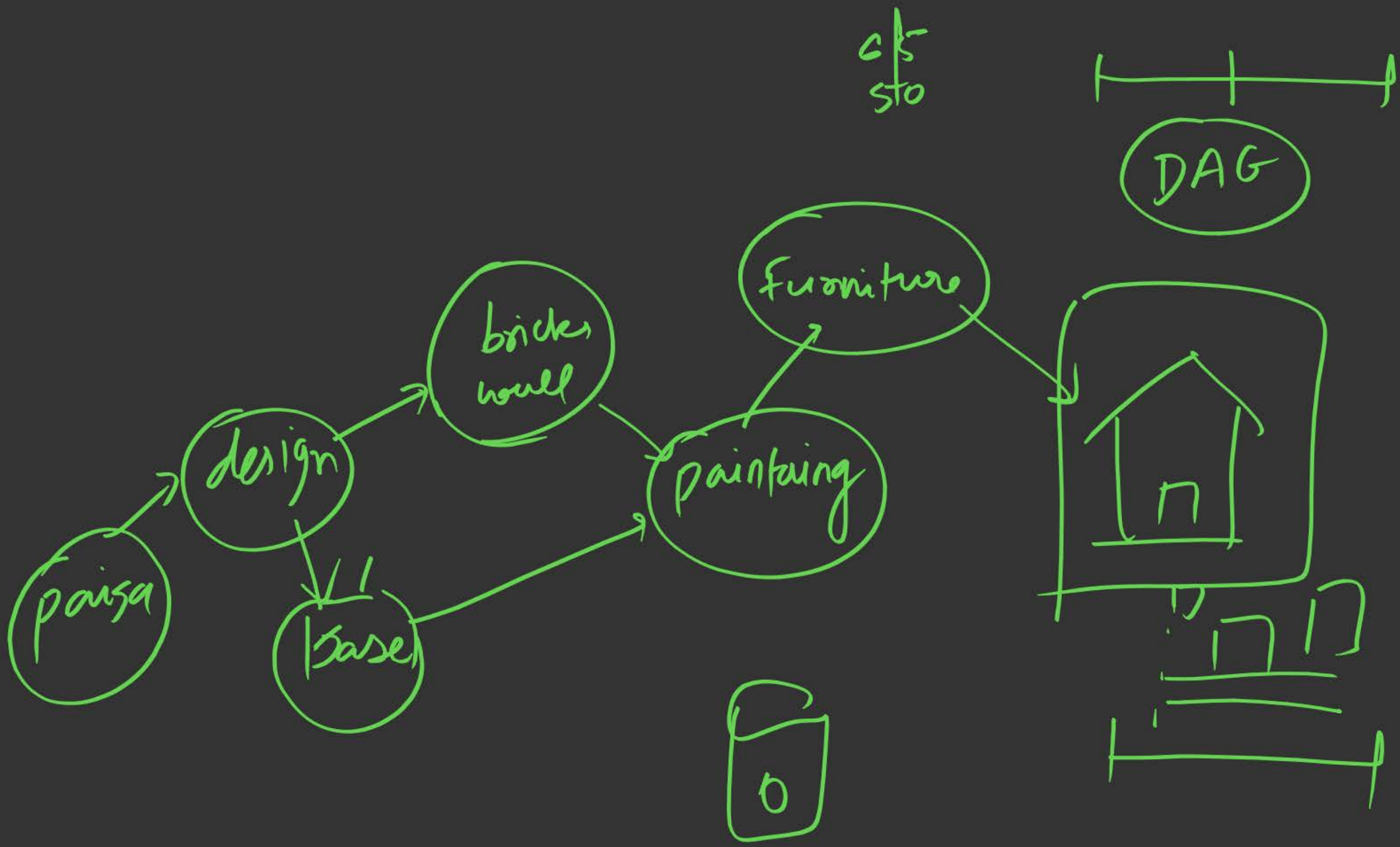
E.g.5.



Cement → Paint

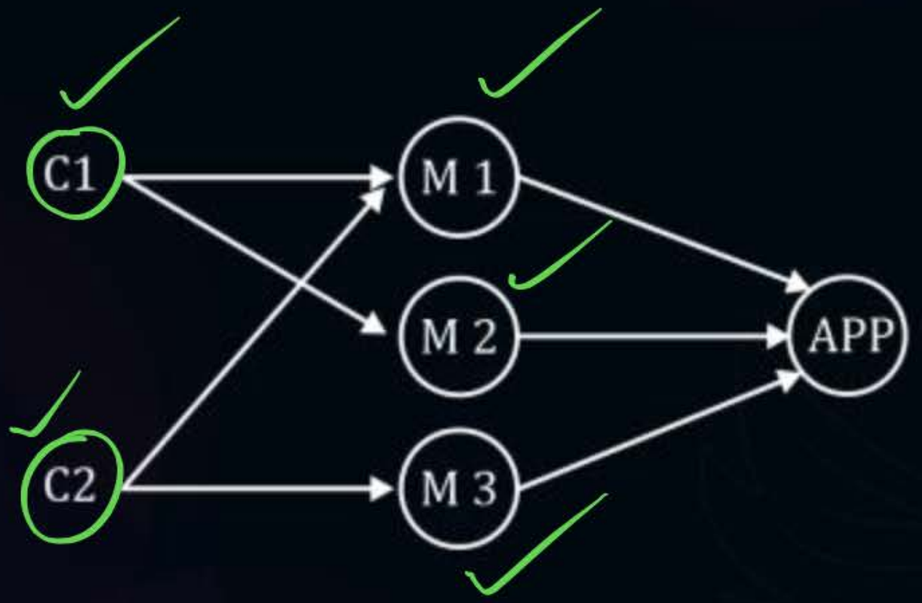
↓

structure


c/s
sto
DAG
furniture
bricks
wall
design
painting
paisa
base
0

Topic : Graph Algorithms

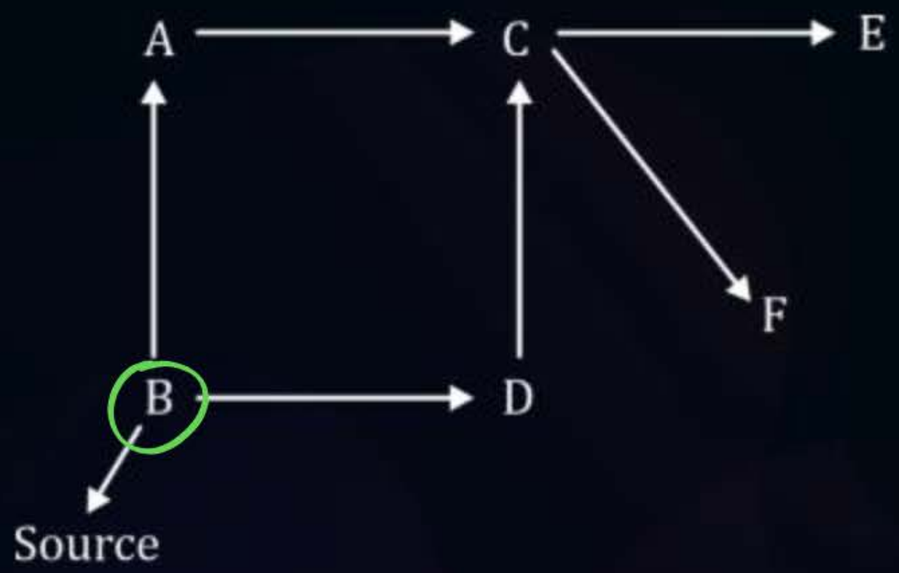
E.g.6.



Topic : Graph Algorithms

E.g.7. Given a DAG

DAG



Sink: Outdegree = 0(no. of outgoing edge)

Source : Indegree = 0 (no. of incoming edges)

Note:- We visit a node only when all of its precedence nodes are visited

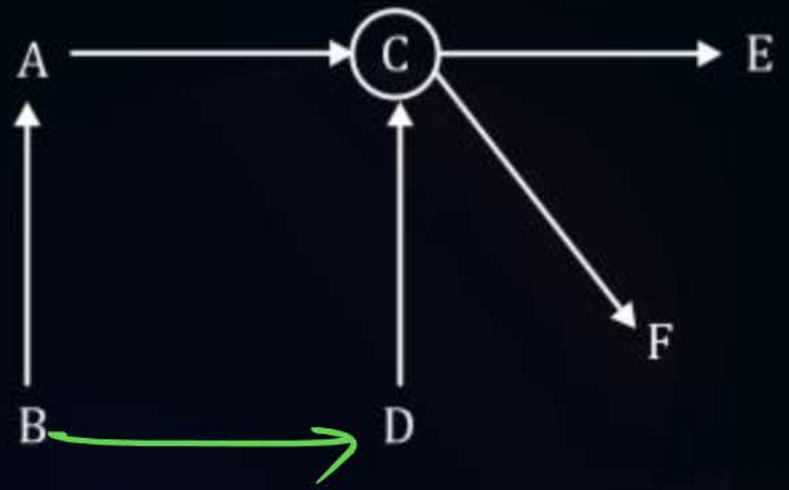
E.g.8. D is precedence of C.

Topological sort

Source→ Sink

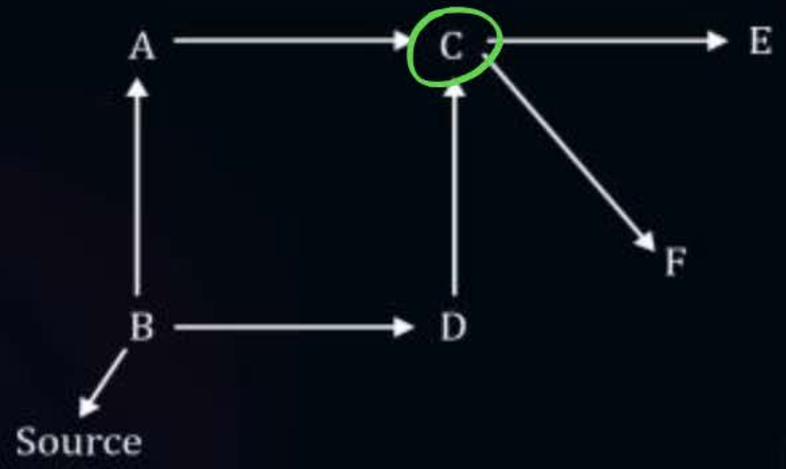
Topic : Graph Algorithms

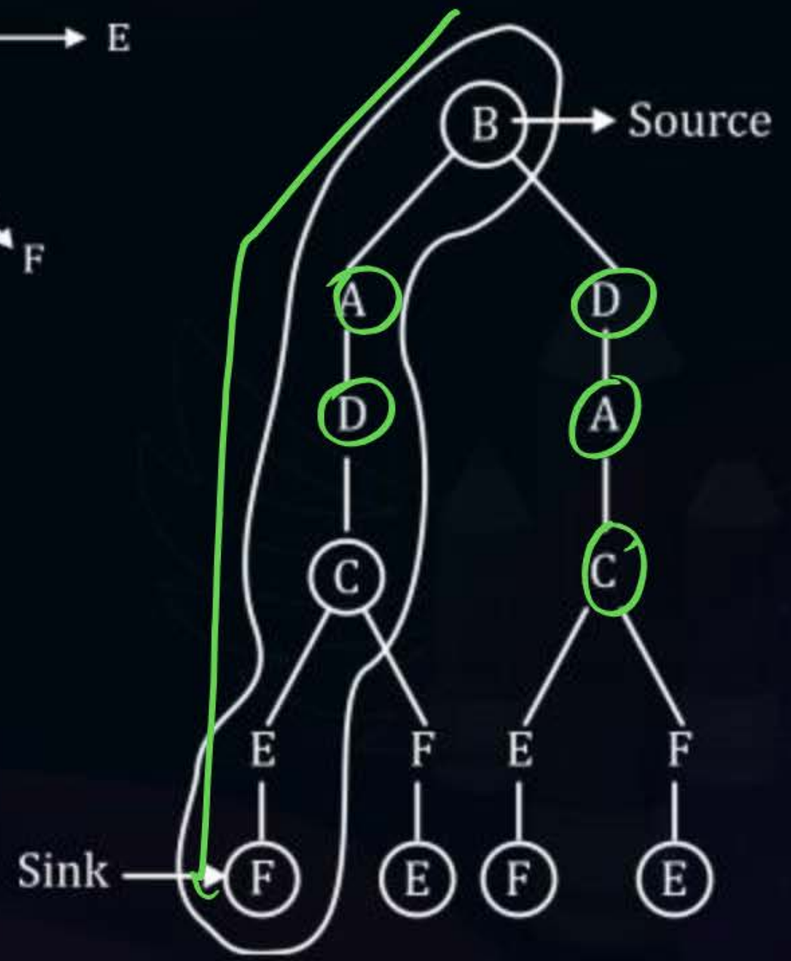
Finding all possible Topological orderings for Given DAG



Topic : Graph Algorithms

#Q. Every branch from source → Sink will give one possible topological order.





A BADCEF

B BADCFE

C BDACEF

D BDACFE

Topic : Graph Algorithms

Algorithm for Topological sort using DFS:

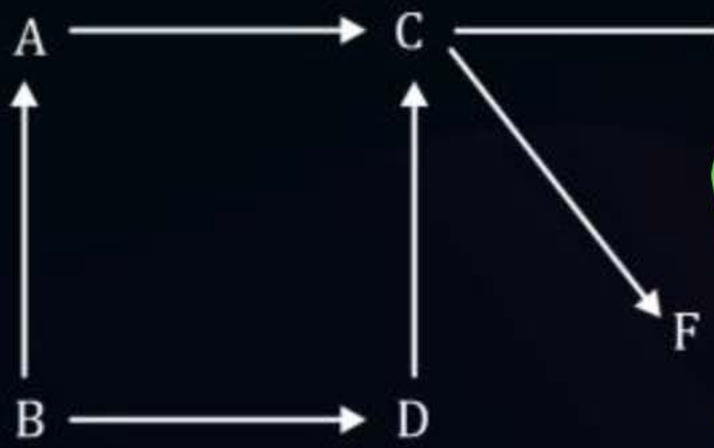
Algo Topological sort (G)

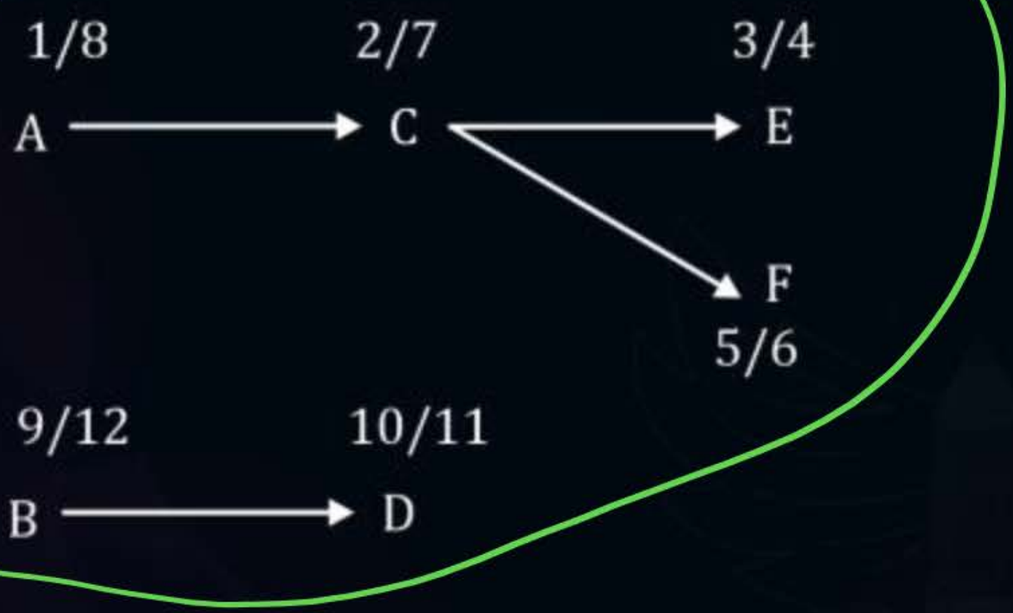
1. Apply DFS staring at any vertex (V): DFS (V)
2. Arrange all the vertices /node obtained in step DFS travels in the descending/decreasing order of their finishing times to get a valid topological sort ordering.

Topic : Graph Algorithms

Case-1: Walkthrough / Dry Run of Prev. Algo

DFS starting at A.





BDACFE

A → 8 ✓
B → 12 ✓
C → 7 ✓
D → 11 ✓
E → 4 ✓
F → 6 ✓

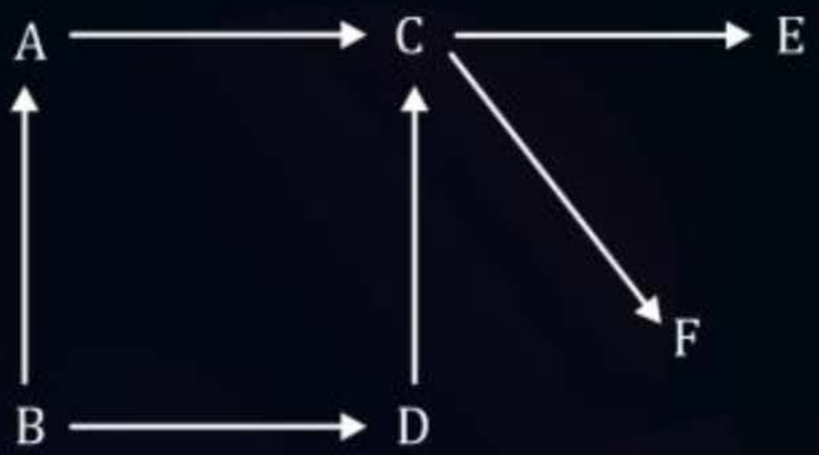
BDACFE

Topic : Graph Algorithms

Case-2:

DFS Starting at E

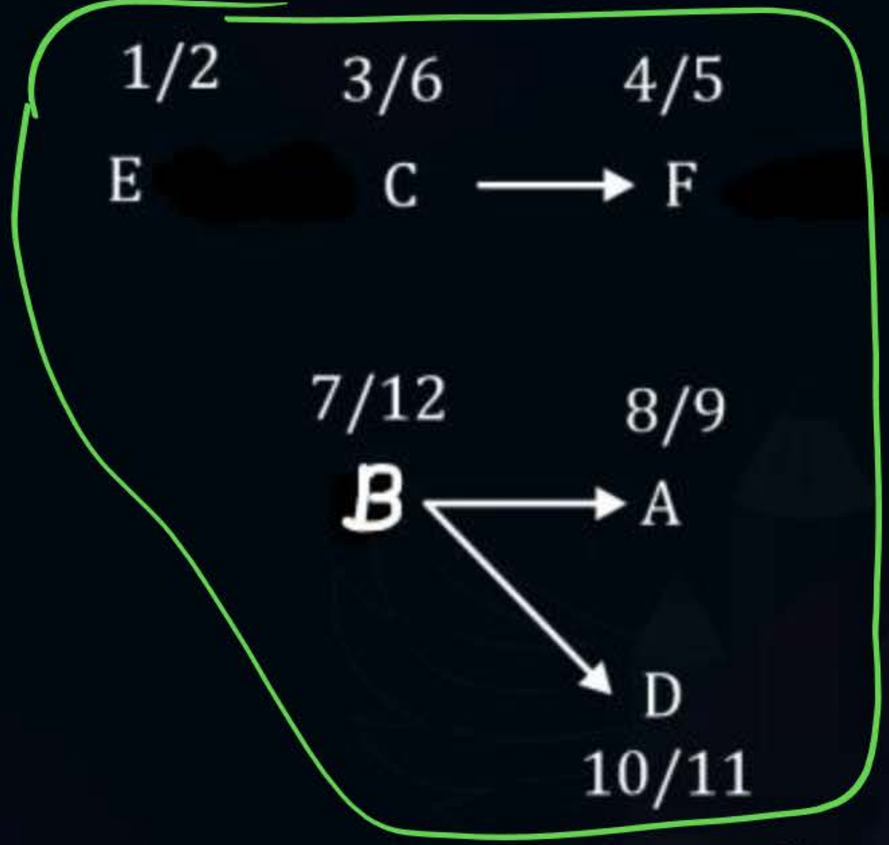
DAG



Finishing times

A→ 9
B→ 12
C→ 6
D → 11
E → 2
F → 5

DFS Spanning forest



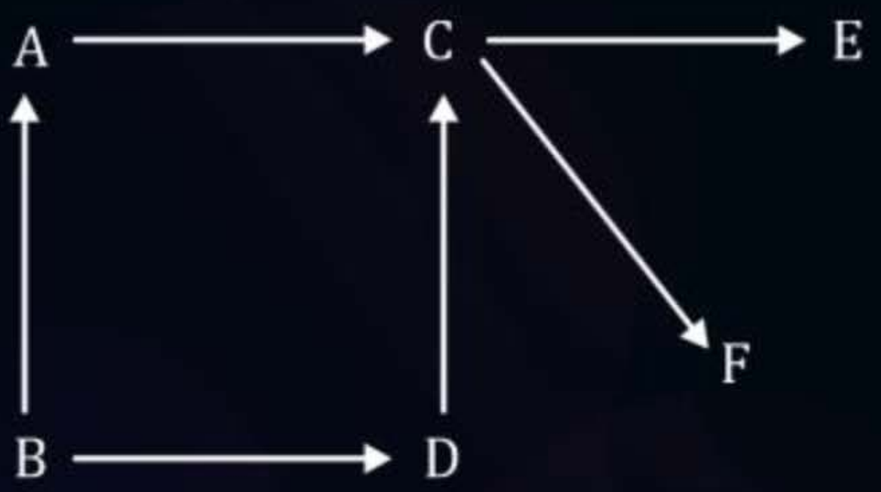
Decreasing order:

BDACFE

Topic : Graph Algorithms

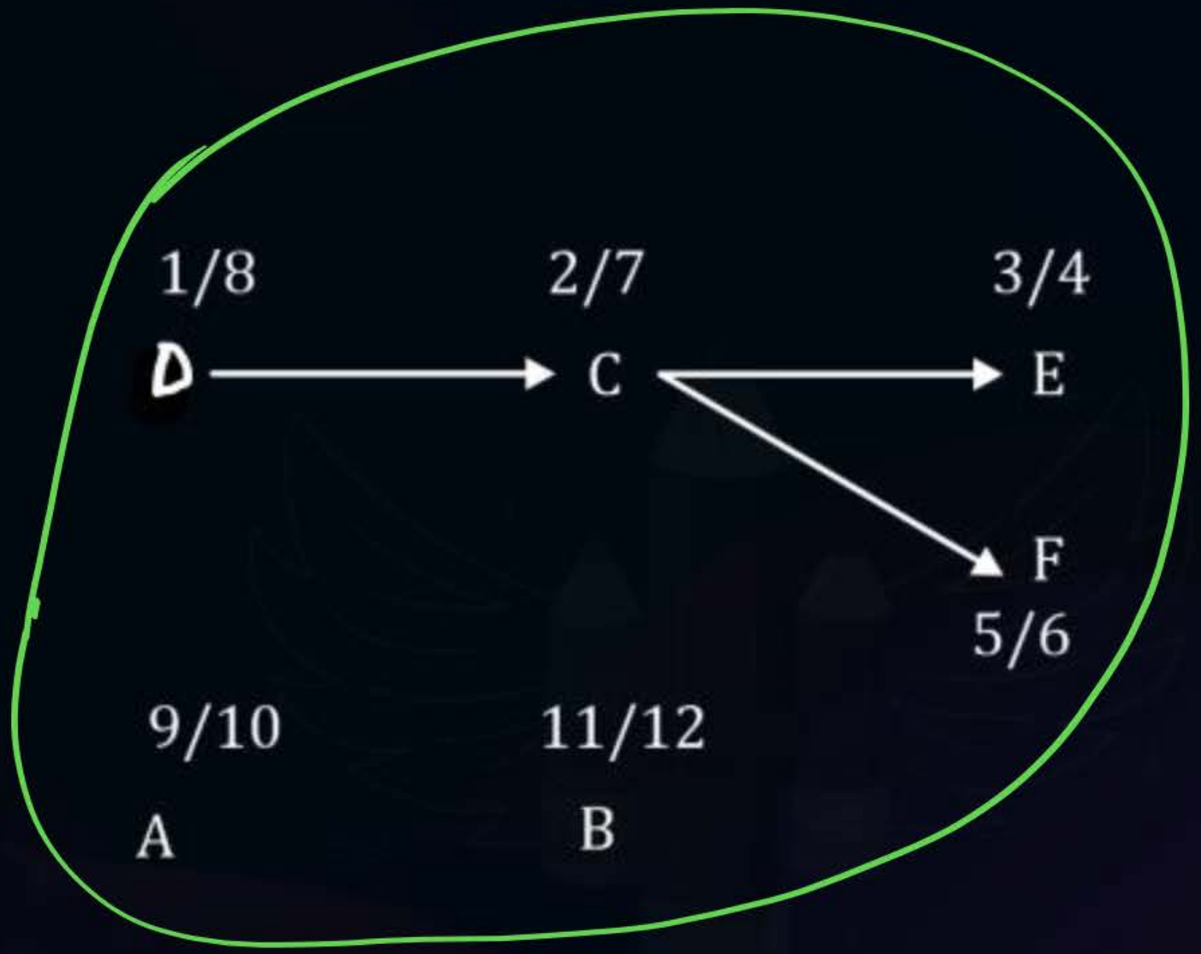
Case-3:

Starting at D:



BADCEF

A-10 D-8
B-12 E-4
C-7 F-6



Topic : Graph Algorithms

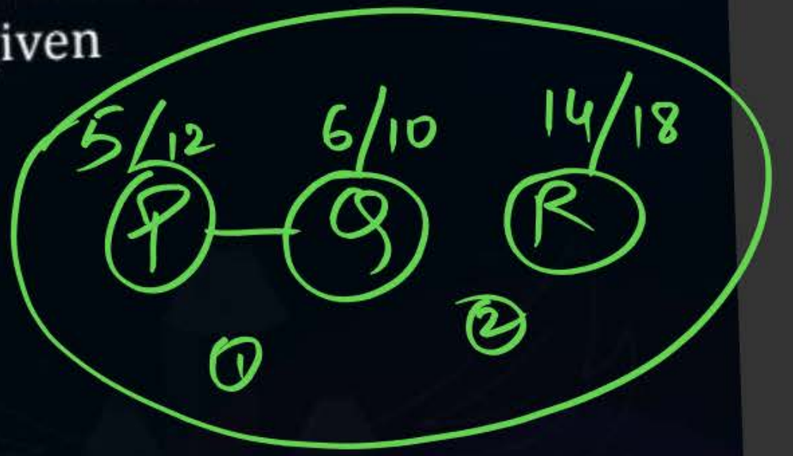
PYQ

100%.

82%.

#Q. Consider the depth-first-search of an undirected graph with 3vertices P, Q, & R. Let discovery time d(V) represent the time instant when the vertex 'V' is visited first, and finish time f(V) represent finishing time ; given

d(p) = 5 f(p) = 12
d(Q) = 6 f(Q) = 10
d(R) = 14 f(R) = 18
Which is true?



A. There is only one connected component.

B. There are two connected components, P and R are connected

C. There are two connected components, Q and R are connected

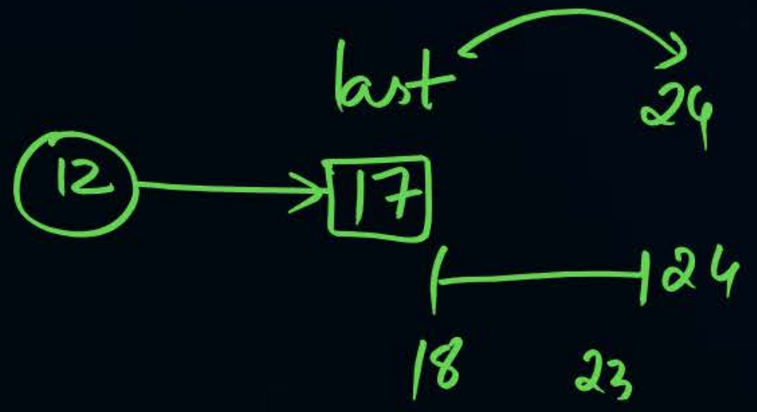
D. There are two connected components, P and Q are connected

2 mins Summary

- Topic
- Topic
- Topic
- Topic
- Topic

DFS on directed

Topological order



Tom
:7:30AM

THANK - YOU