CS & IT ENGINEERING

Algorithms

Graph Algorithms

Lecture No.- 01

By- Aditya Jain sir

Recap of Previous Lecture

Topic

Topic

Dynamic Programming.

Topics to be Covered

- Topic
- Topic
- Topic

Graph Algorithms

↳ Traversals

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

Topic : Graph Traversals

Graph Traversals:

Graph traversal is a process of visiting each and every node of the tree/graph in some specific order and processing the information exactly once.

Topic : Graph Traversals

Graph Traversals:

(1) Depth First Search (DFS)/ Traversal (DFT)

(a) Undirected graph

I. Connected

II. Disconnected

(b) Direct graph

I. Directed acyclic graph (DAG)

II. Topological Sorting / ordering

Topic : Graph Traversals

Graph Traversals:

(2) Breadth First Search (BFS)/Traversal (BFT)

(a) FIFO BFS (Default / standard Approach)

(b) LIFO BFS (Bonus)

Topic : Graph Traversals

DFS ⟶ Stack

BFS ⟶ Queue

Stack → LIFO

Queue → FIFO

Topic : Graph Traversals

Applications of DFS and BFS:

→ Topological Sorting

→ Connected Components

→ Strongly Connected Components

→ Articulation point / cut vertex

→ Biconnected Components

BFS

Topic : Graph Traversals

(1) Breadth First Traversal (BFT)

(a) FIFO BFS/BFT (default)

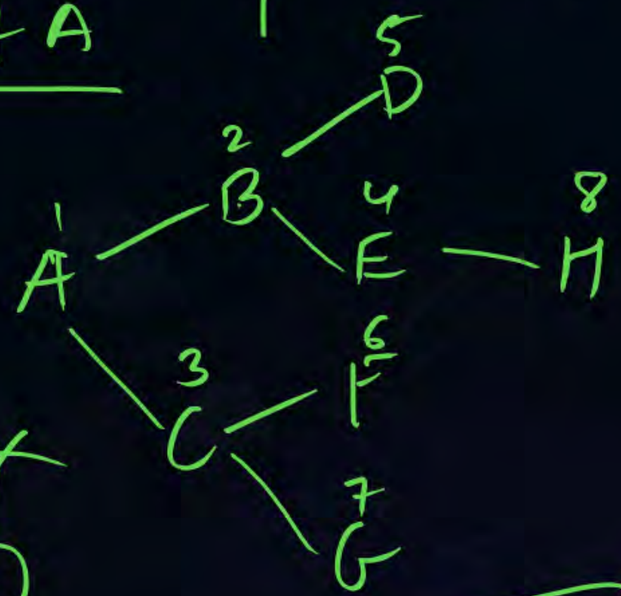
e.g. Given

BFS → Starting at A

Queue FIFO

Node	A	B	C	E	D	F	G	H
Parent	-	A	A	B	B	C	C	E

BFS Spanning Tree (ST)



BFS: ABCEDFGH

Starting with A Vertex

BFS Spanning Tree

Queue Node	B	C	D	E	G	F	H
parent	A	A	B	B	C	C	D

Output: ABCDEGFH

Topic : Graph Traversals

Observations:

Assume: every edge cost = 1

A ⟶ D

1. A – B – D ⟶ 1 + 1 = 2 cost
2. A – B – E – H – D ⟶ 1 + 1 + 1 + 1 = 4 Cost

Topic : Graph Traversals

Property:

Or unweighted graph

In a unweighted graph (where every edge has equal weight)

BFS solves the problem of (SSSP) single source shortest paths from starting vertex to all other vertex.

Topic : Graph Traversals

Bonus:

diff from DFS

LIFO BFS (Special case)

There we are a Queue that follows LIFO order (Last in first out)

Start

BFS ST

Queue (LIFO)

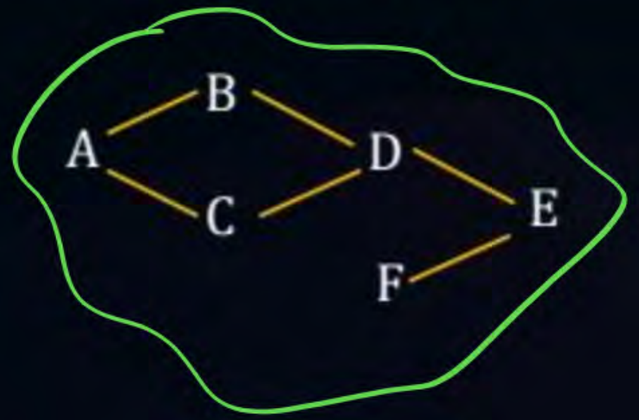
Node	B	C	F	G	H	D	E
parent	A	A	C	C	G	H	H

output: ACGHEDFB

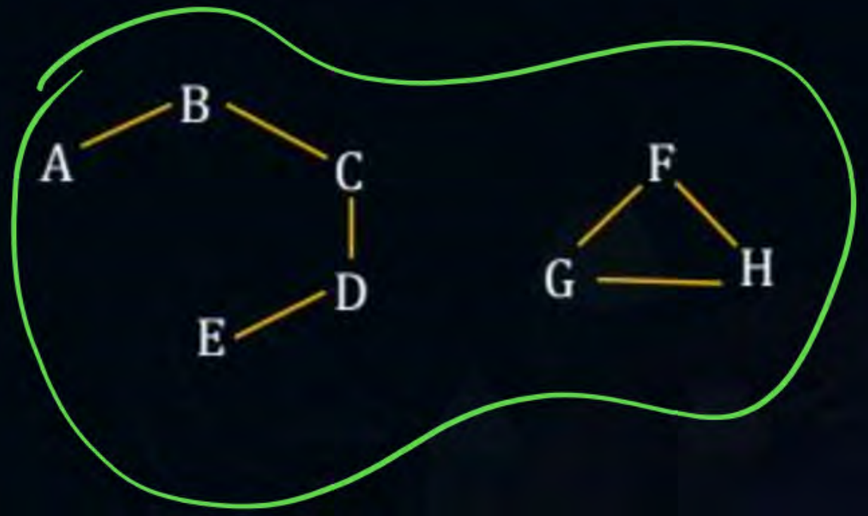
Topic : Graph Traversals

Depth First Traversals (DFT/DFS)

1. Connected graph



(2) Disconnected graph



Topic : Graph Traversals

Depth First Traversal (DFT)

(a) DFS/DFT on an undirected Graph; (Connected Graph)

I. E-node (Exploring node)

→ Node which is currently getting explored.

II. Live Node.

→ Node that is NOT yet fully explored. (live nodes are present in stack)

III. Dead Node.

→ Node that has already been fully explored (already popped out from stack).

Topic : Graph Traversals

V.V.V. imp

During Traversal, the different types of timing values that a node is associated with.

(1) Discovery time: d(n)

→ It is the time at which the node has got discovered/visited for the very first time.

(2) Finishing time: t(n)

→ It is the time at which the node becomes a dead node.

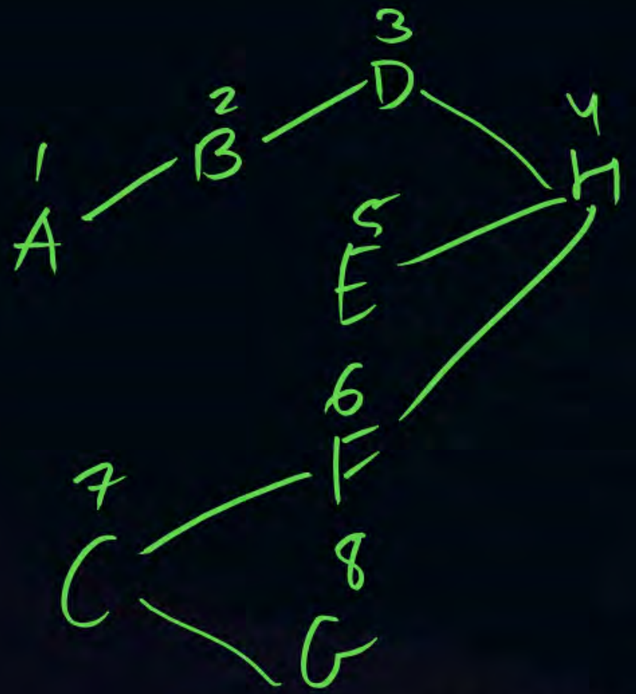
d/f

A

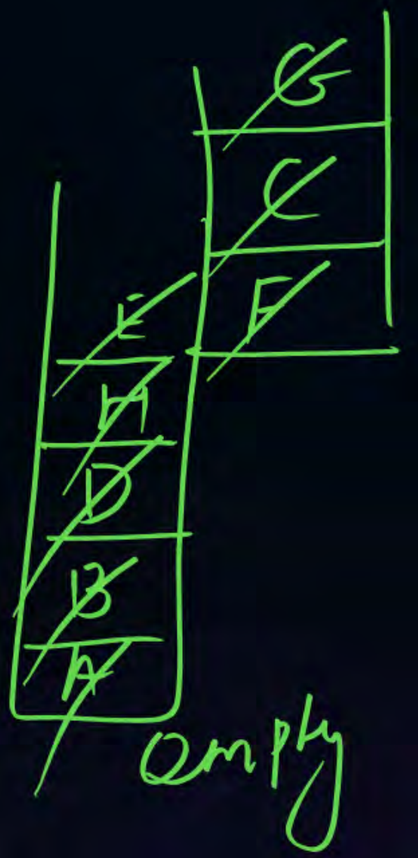
Topic : Graph Traversals

DFS in undirected connected graph:

Start



Stack



output: ABDHEFCG

Stack empty ⟶ **End of DFS**

Multiple BFS, DFS Sequence possible for a graph

ACGHDBEF ⟶ also valid DFS starting from A.

Topic : Graph Traversals

#Q. Which of the following represent valid DFS traversal on graph G?

(2) BFS?

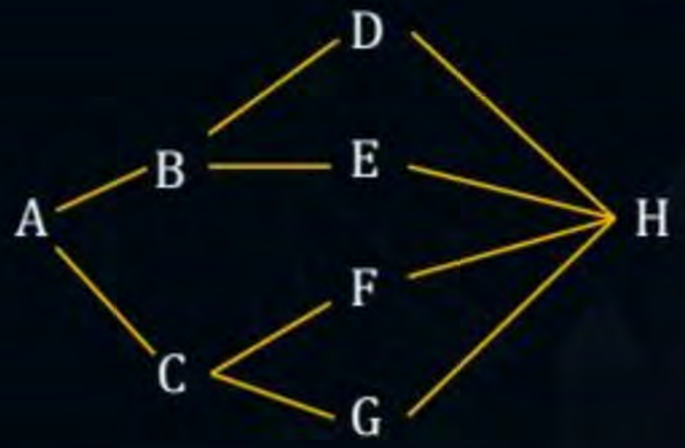
A. A B F E D H G C

B. A B E H D F C G

C. D H G C F A B E

D. H E B D A C G F

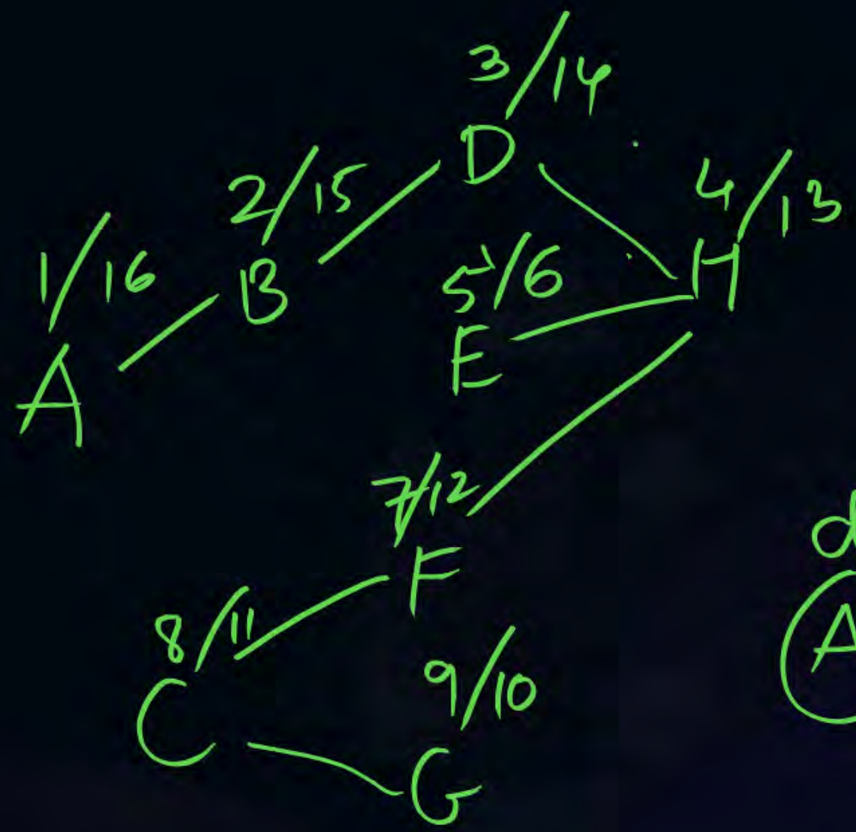
E. H D B E F C G A



A X
B X
C X
D X
E ✓

(E) None

A ABFEDHGC



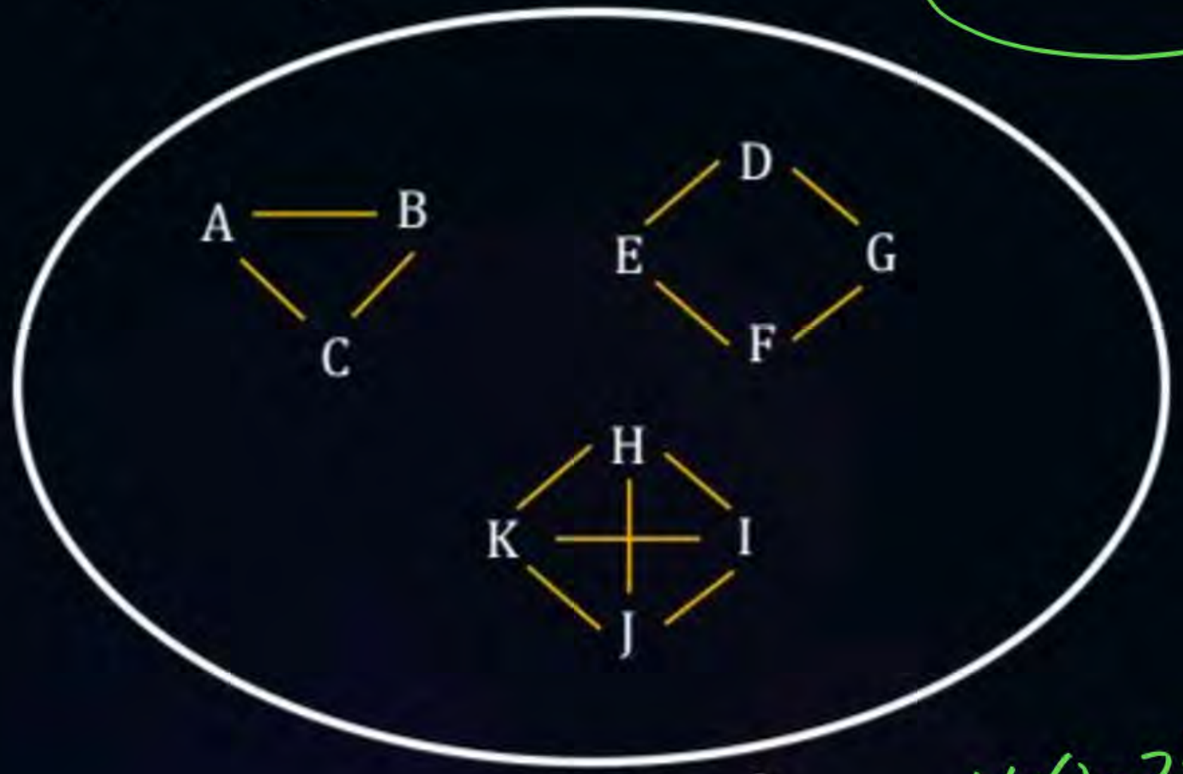
Shortcut : 2 × 8 = 16

d/f

A

Topic : Graph Traversals

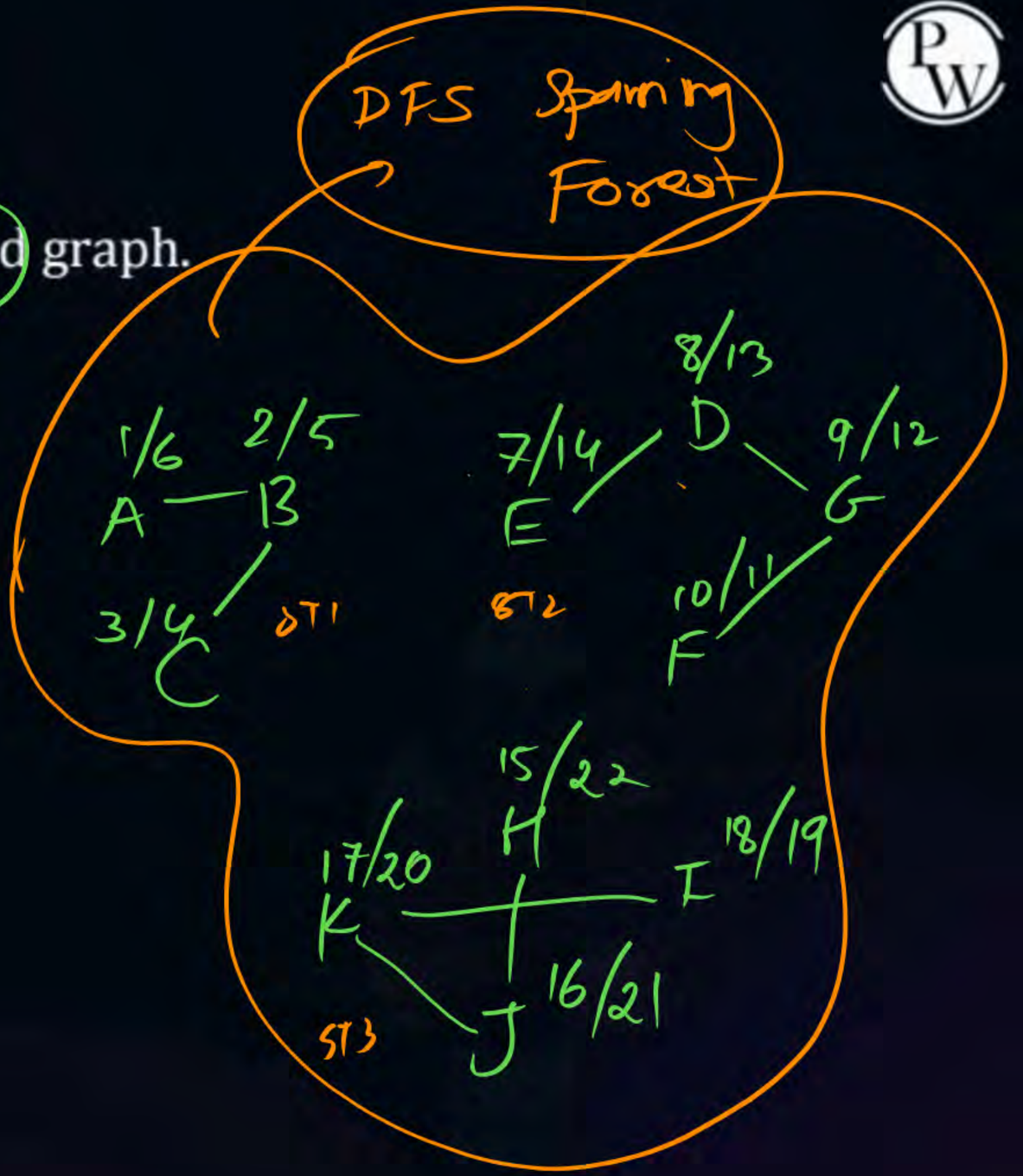
#Q. DFS / DFT on a undirected Disconnected graph.



G(V, E)

11 X 2 = 22

DFS Spanning Forest



Topic : Graph Traversals

V.V.Imp Application of discovery & Finishing time

#Q. You are given a graph (G) with 5 vertices {A, B, C, D, E}
The Discovery and Finishing times for each node/vertex after carrying out DFS are as follows.

Topic : Graph Traversals

10 → 2

#Q. You are given a graph (G) with 5 vertices {A, B, C, D, E}
The Discovery and Finishing times for each node/vertex after carrying out DFS are as follows.

1. How many connected components are in G?

2. What are the vertices in each of those connected components for each question.

(d, f)	A	B	C	D	E	
1.	(1, 2)	(3, 5)	(7, 8)	(11, 12)	(18, 20)	→ 5 Comp {A} {B} {C} (D) (E)
2.	(1, 15)	(3, 13)	(4, 12)	(5, 11)	(6, 9)	→ (A – B – C – D – E) 1 Comp
3.	(1, 14)	(2, 7)	(16, 18)	(19, 25)	(21, 22)	→ (A – B) (C) (D – E) 3
4.	(1, 16)	(3, 11)	(5, 10)	(20, 25)	(22, 23)	→ (A – B – C) (D – E) 2

1 single node is also a connected component.

NOTE: In D/F times, its only mandatory to have time in increasing order

5 10 12 18

2 mins Summary

- Topic
- Topic
- Topic
- Topic
- Topic

Graph Traversals

BFS

DFS — UC/UDc

THANK - YOU