DS & AI ENGINEERING

Artificial Intelligence

Un-Informed search

Lecture No.- ☎ 03

By- Aditya sir

Recap of Previous Lecture

- Topic
- Topic — BFS
- Topic — DFS on Tree
- Topic

Topics to be Covered

- Topic
- Topic
- Topic

DFS on Graphs

Practice

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 in ML
8. Completed my Masters with an overall GPA of 9.36/10
9. Joined Dream11 as a Data Scientist
10. Have mentored 15,000+ 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 : Analysis of Un-Informed Search

DFS on Graphs

Topic : Uninformed Search

BFS:

- Queue → FIFO.
- Scan complete level in on go.
- 2 List
  - Queue (open)
  - Visited nodes. (closed)

Topic : Uninformed Search

Breath first search

- Breadth-first search (BFS) is an algorithm for traversing or searching tree or graph data structures.
- It starts at the tree root (or some arbitrary node of a graph, sometimes referred to as a 'search key'), and explores all of the neighbor nodes at the present depth prior to moving on to the nodes at the next depth level.
- It is implemented using a queue (FIFO).
- Breadth First Search (BFS) is a graph traversal algorithm that explores all the vertices in a graph at the current depth before moving on to the vertices at the next depth level. It starts at a specified vertex and visits all its neighbors before moving on to the next level of neighbors. BFS is commonly used in algorithms for pathfinding, connected components, and shortest path problems in graphs.

Topic : Uninformed Search

Breath first search

- The Breadth-First Search is a traversing algorithm used to satisfy a given property by searching the tree or graph data structure.
- It belongs to uninformed or blind search AI algorithms as It operates solely based on the connectivity of nodes and doesn't 'prioritize' any particular path over another based on heuristic knowledge or domain-specific information.
- It doesn't incorporate any additional information beyond the structure of the search space. It is optimal for unweighted graphs and is particularly suitable when all actions have the same cost. Due to its systematic search strategy, BFS can efficiently explore even infinite state spaces

Topic : Uninformed Search

BFS Algo:

→ If the solution exist then BFS will surely find goal state in state space

→ Completeness property of Algo.

BFS → 1) Complete

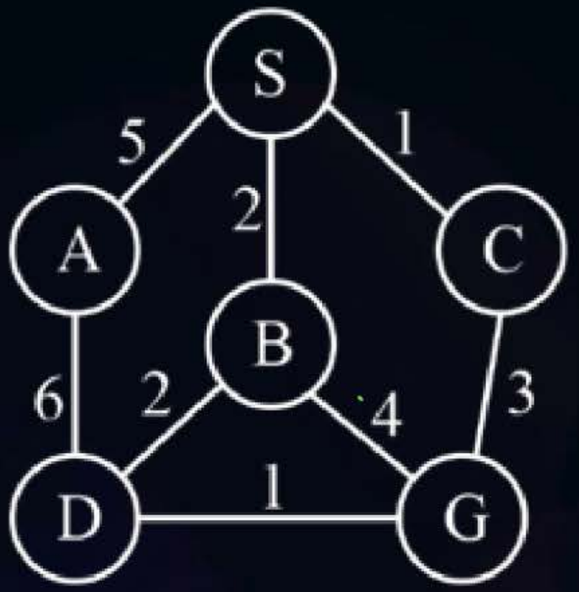
BFS → 2) optimal

DFS	BFS	DFS
a) Node is visited (present in closed list)	X	X
b) Node not yet visited but is present in open list	no need to write again in open list	write again in open list

not in open list again

Topic : Uninformed Search

BFS
S → G



Alphabetical

no. of edges

S → G

BFS: len of path S → G ?

S → B → G

open

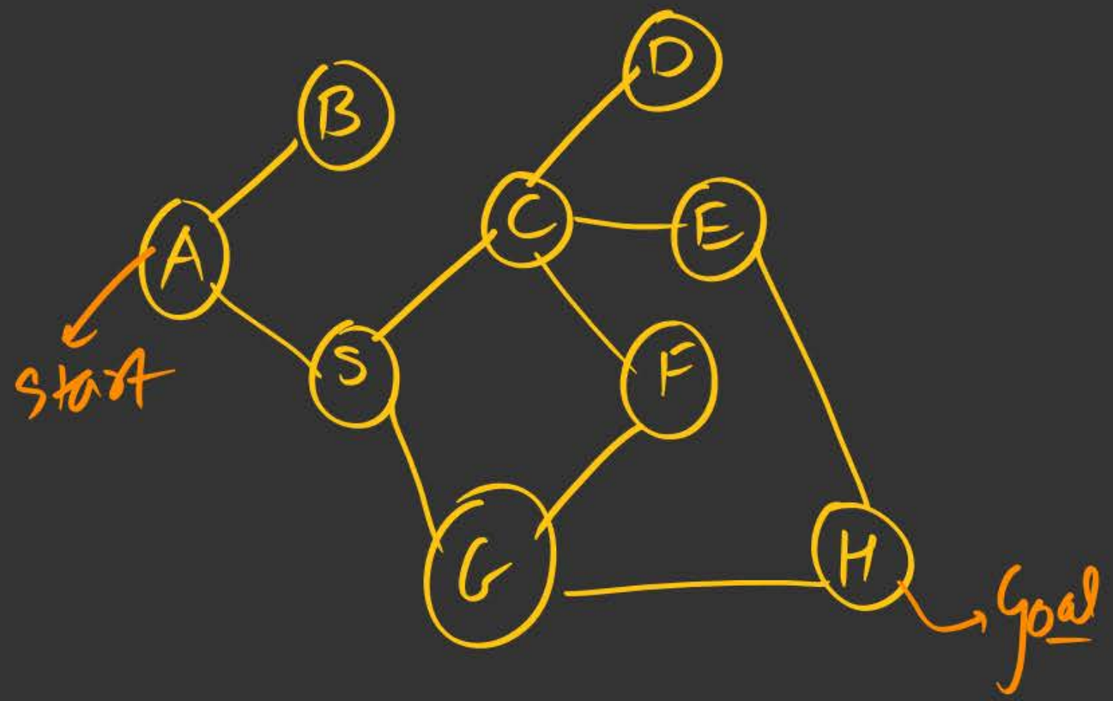
~~S~~ ~~A~~ ~~B~~ ~~C~~ ~~D~~ ~~G~~

S B

closed

SABCDG

stop



A → H : A → S → G → H

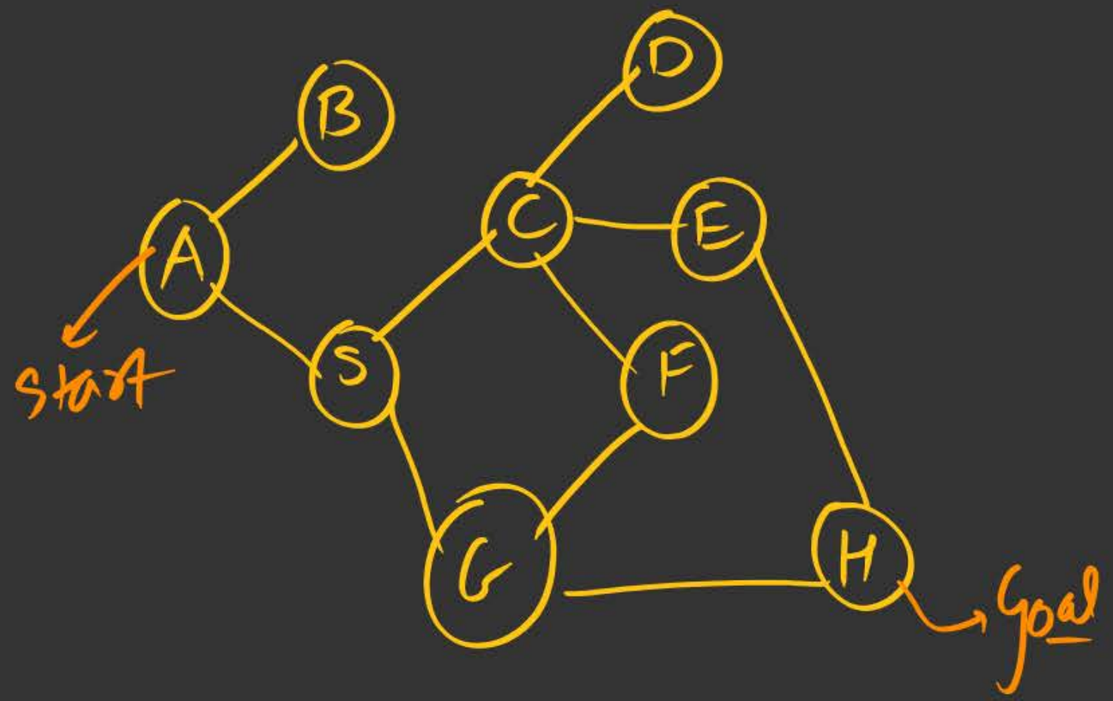
open (FIFO)

A B S C G D E F H

Closed

A B S C G D E F H

goal



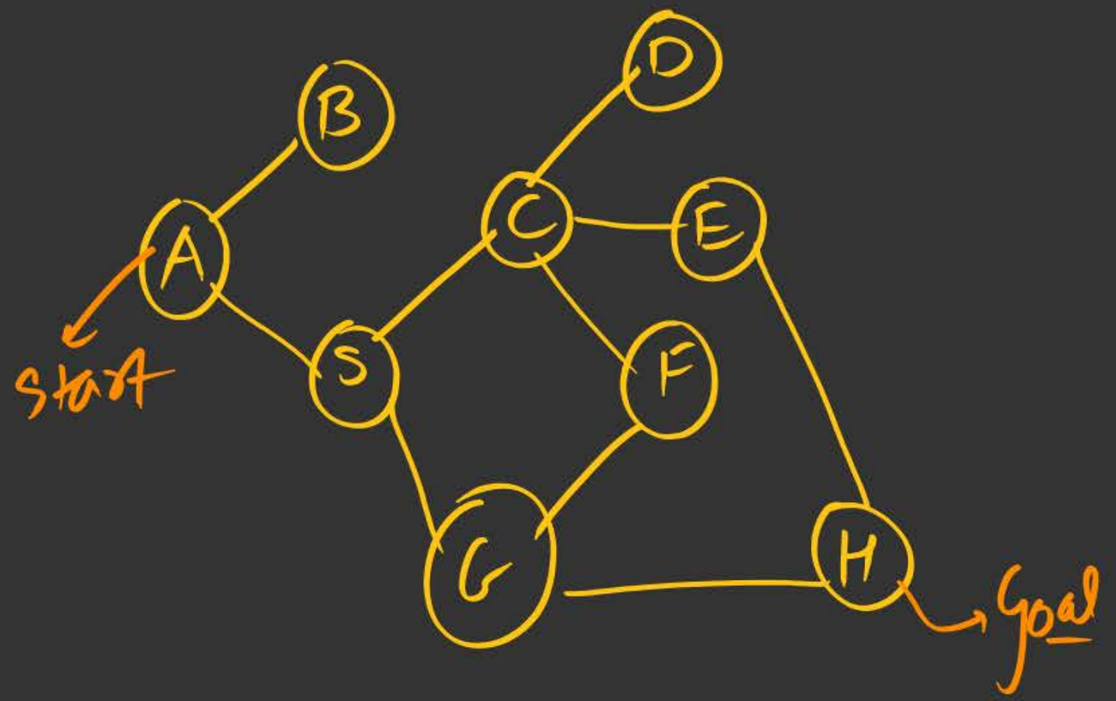
Open (LIFO)

~~A~~ ~~S~~ ~~B~~ G ~~C~~ F ~~E~~ ~~D~~ ~~H~~ G

closed

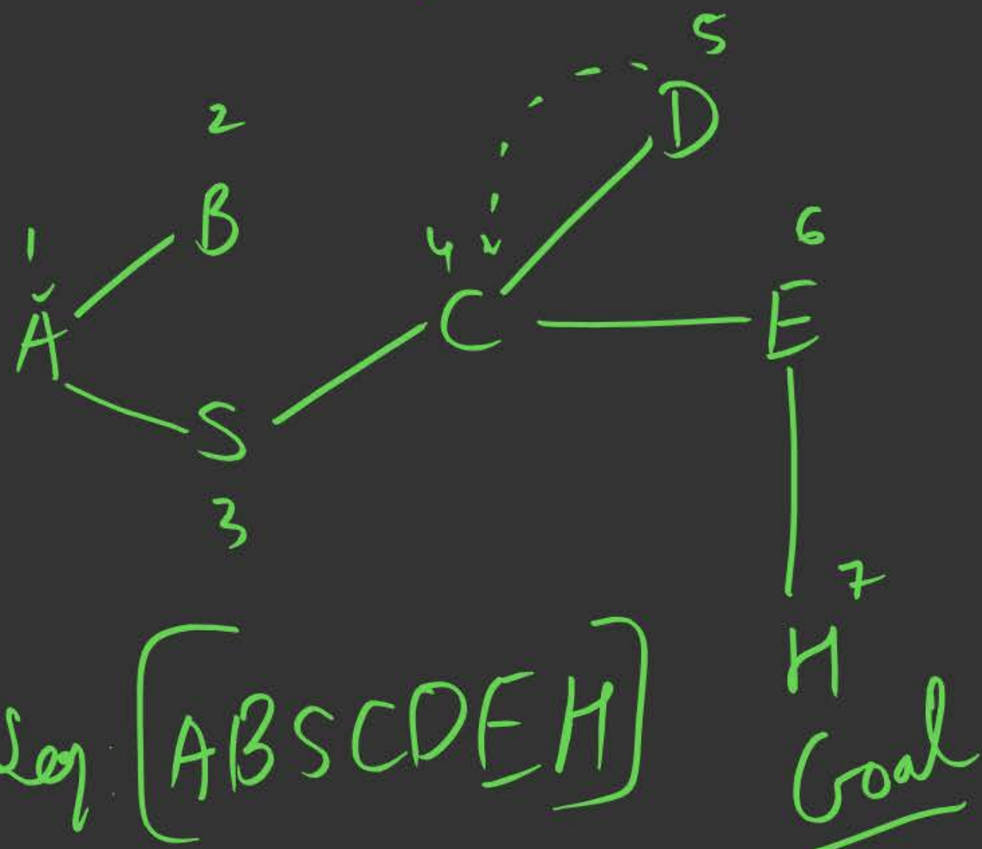
A B S C D E H

Goal

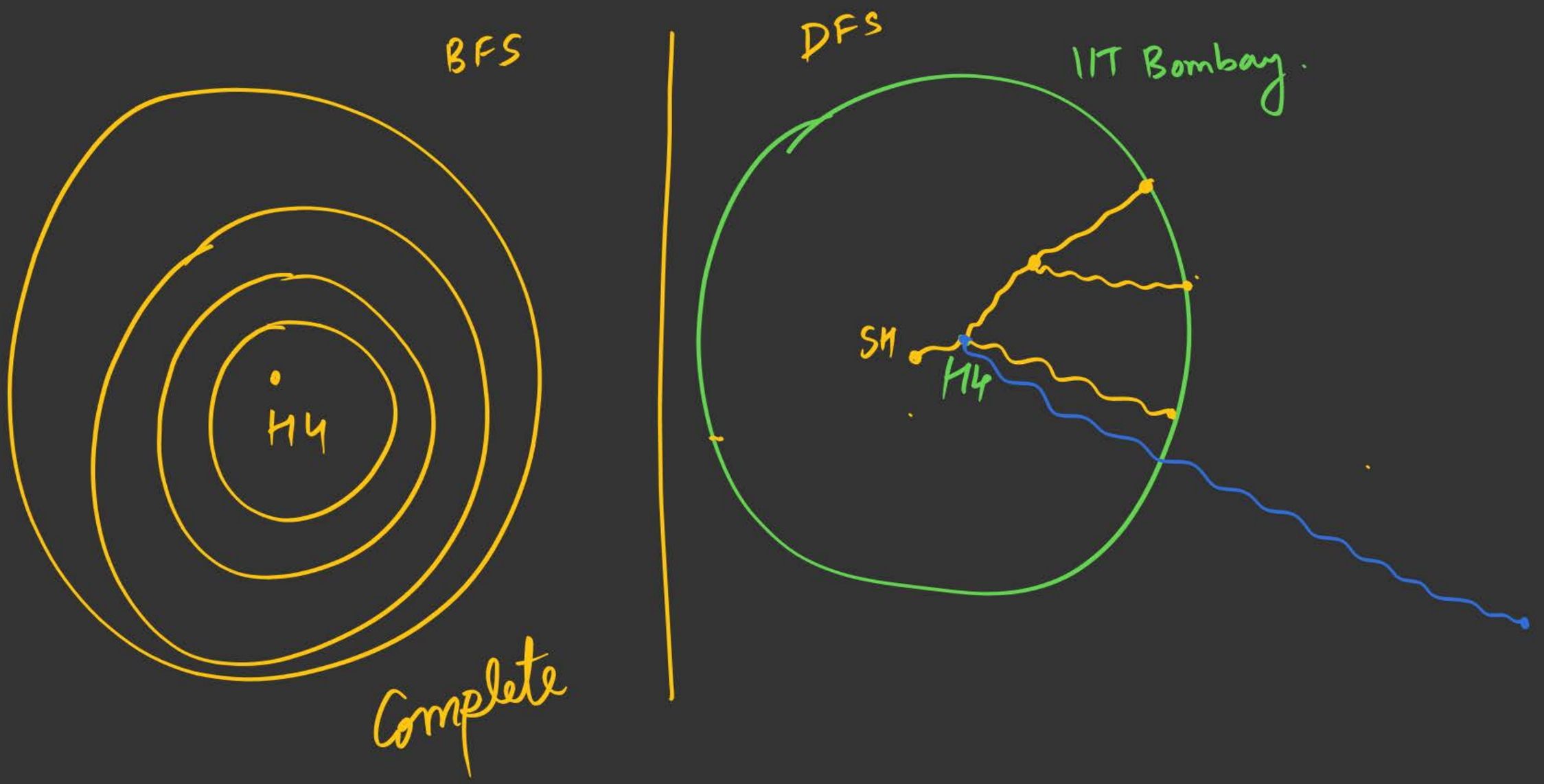


DFS Spanning Tree

(Algo)



Visited Seq [ABSCDEH]


BFS
HY
Complete
DFS
IIT Bombay.
SM
HY

* Properties of DFS

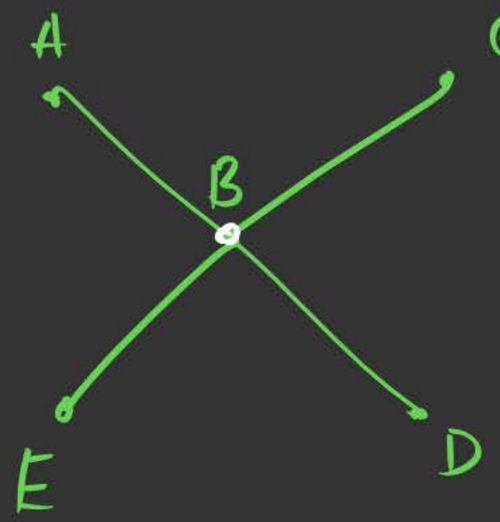
↳ 1) Not optimal search algo

Not Complete ←

2) when Search Space is infinite, then it might not give soln, even if it exists.

(Q) How many diff possible BFS orderings?

7.5%



(A) B 3 × 2 × 1

Start at A → 6
" C → 6
D → 6
E → 6

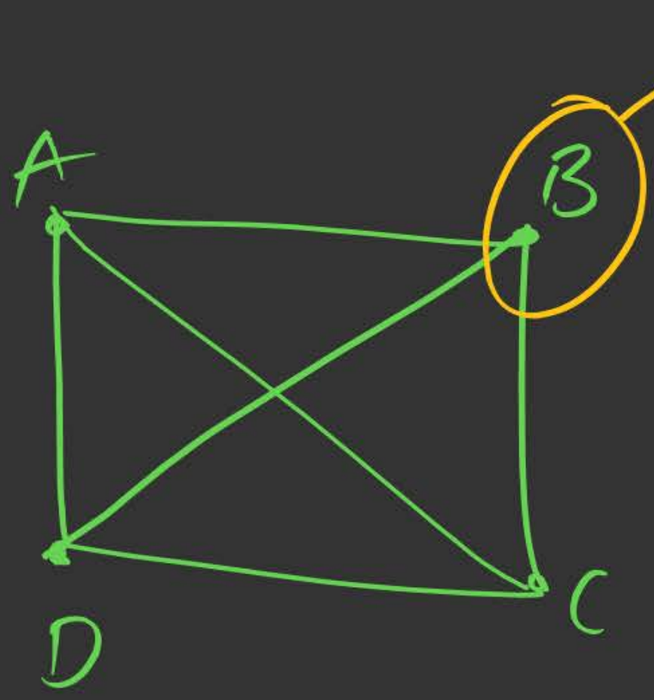
6 × 4 = 24

Total orderings
= 24 + 24
= 48 ✓

Start at B: B 4 × 3 × 2 × 1 ⇒ 24

(Q.2) Given a Complete graph with 4 vertices.

How many diff BFS orderings are possible from a given starting node?



Start

fix

B 3 × 2 × 1

= 6

Part 2 :- Starting node not fixed.

→ 24

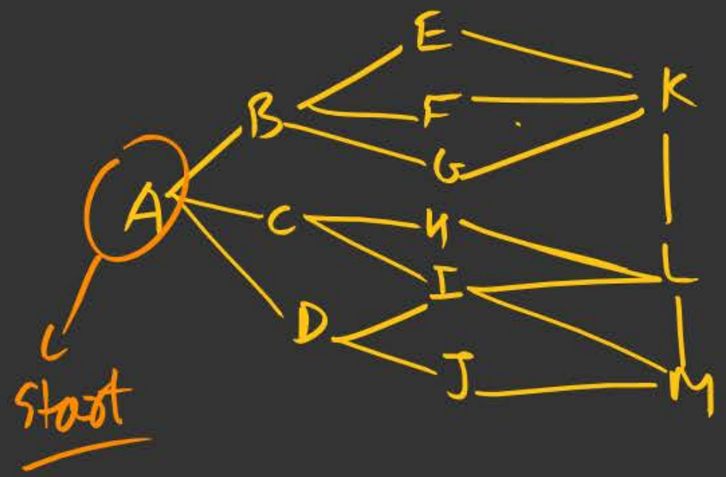
In General :- Complete Graph (K_n)

No. of diff BFS orderings = $\boxed{n!}$

$n = 4 \longrightarrow 4! = 24$

$n = 5 \longrightarrow 5! = 120$

(Q) Alphabetical



Goal: G and I.

1) BFS → G ✓

A → G : len(path)

A → B → G

↳ len = 2

Open

~~A~~ ~~B~~ ~~C~~ ~~D~~ ~~E~~ ~~F~~ ~~G~~ H I J K

Closed

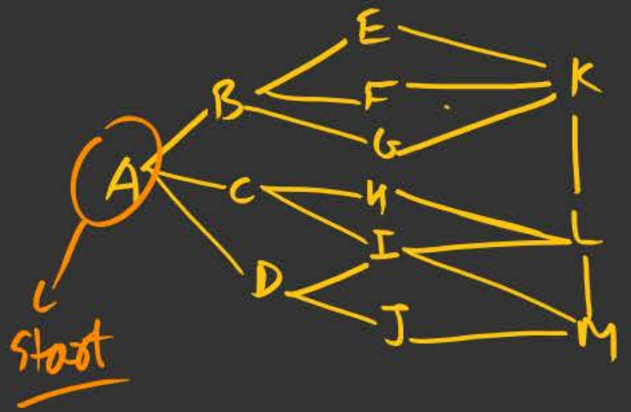
A B C E F G

↳ Stop

Goal: G & I

② DFS : G ✓

(Q) Alphabetical



DFS: A → G

open

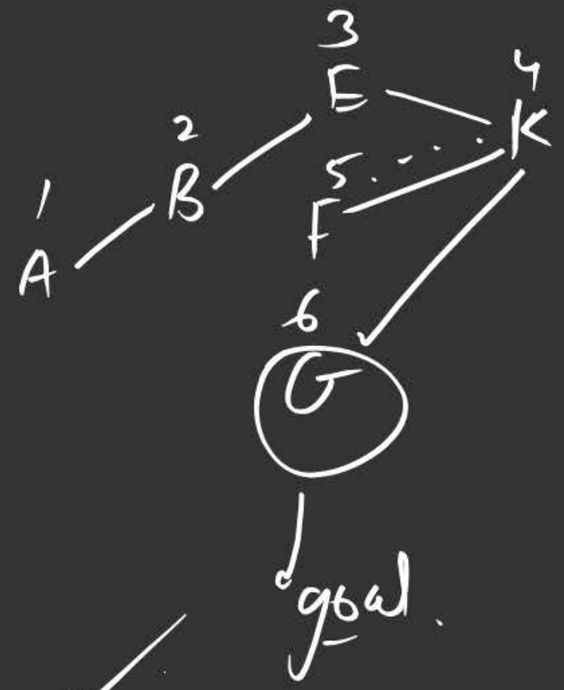
~~A~~ D C ~~B~~ G F ~~E~~ ~~K~~ L ~~G~~ ~~F~~

Closed

A B E K F G

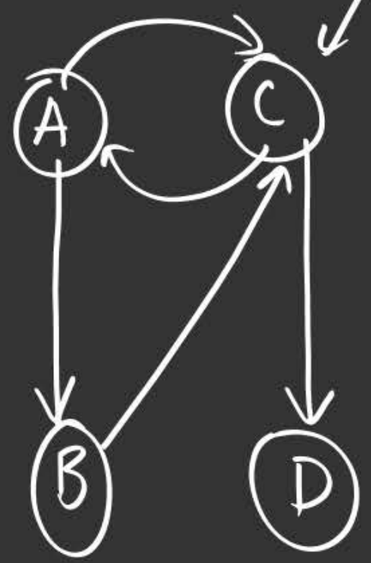
↳ goal

Algo logic



A B E K F G

(Q)



Which is correct BFS?

A) CADB

B) CDAB

C) CBAD ✗

D) CDBA ✗

A) CADB

B) CDAB

C) CBAD ✗

D) CDBA ✗

THANK - YOU