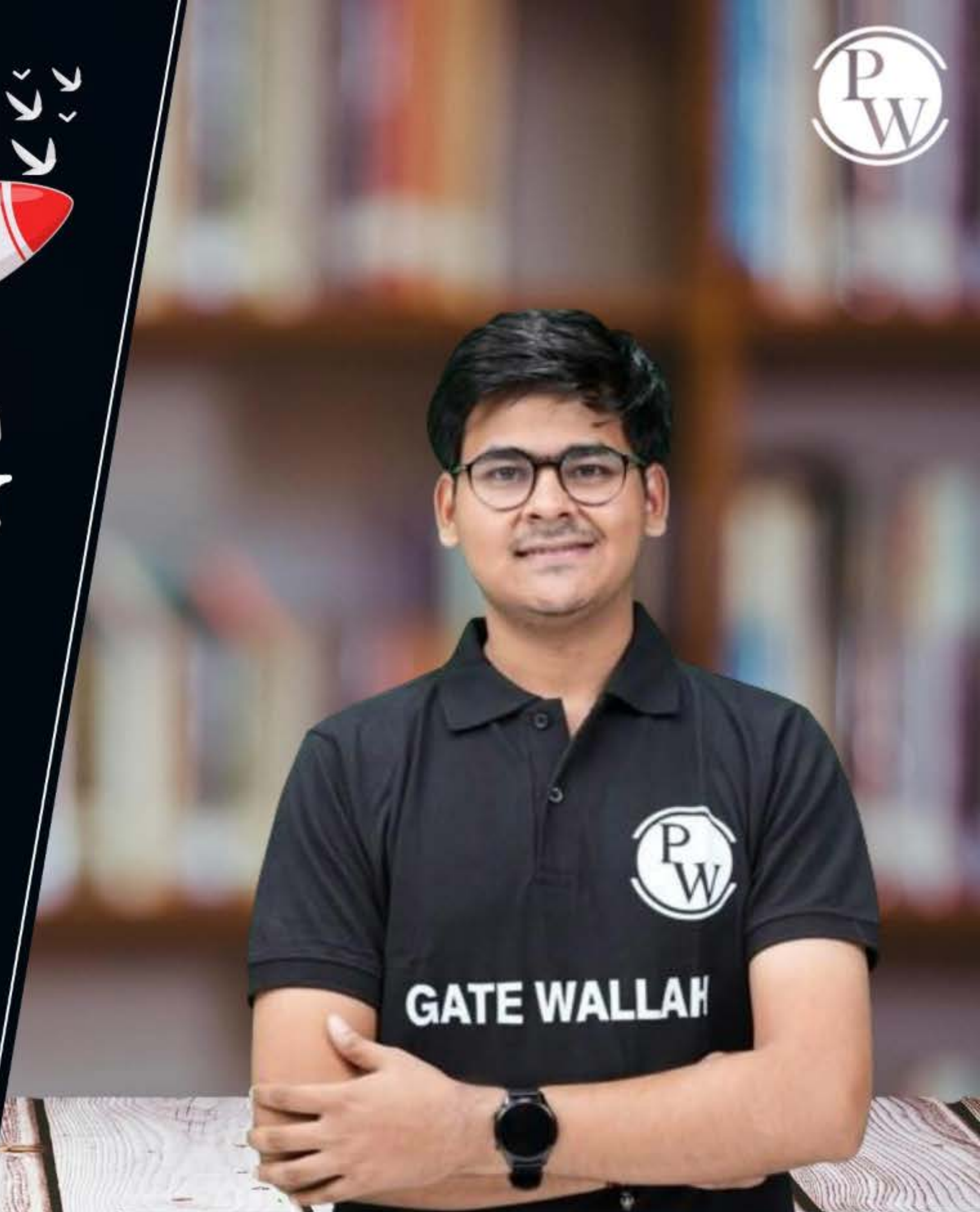
DS & AI ENGINEERING

Artificial Intelligence

Un-Informed search

Lecture No.- 05



By- Aditya sir

Recap of Previous Lecture

Topic

Topic DFS

Topic BFS

Topic DLS

Topics to be Covered

Topic

Topic Questions

Topic IDDFS

UCS

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

@ADITYASIR_PW

Telegram Link for Aditya Jain sir:

https://t.me/AdityaSir_PW

Topic : Uninformed Search

DFS $\Rightarrow$ tree

$A \rightarrow H$

$\Rightarrow$(Alphabetical order)

1) BFS

Path: A→B→C→H

72%

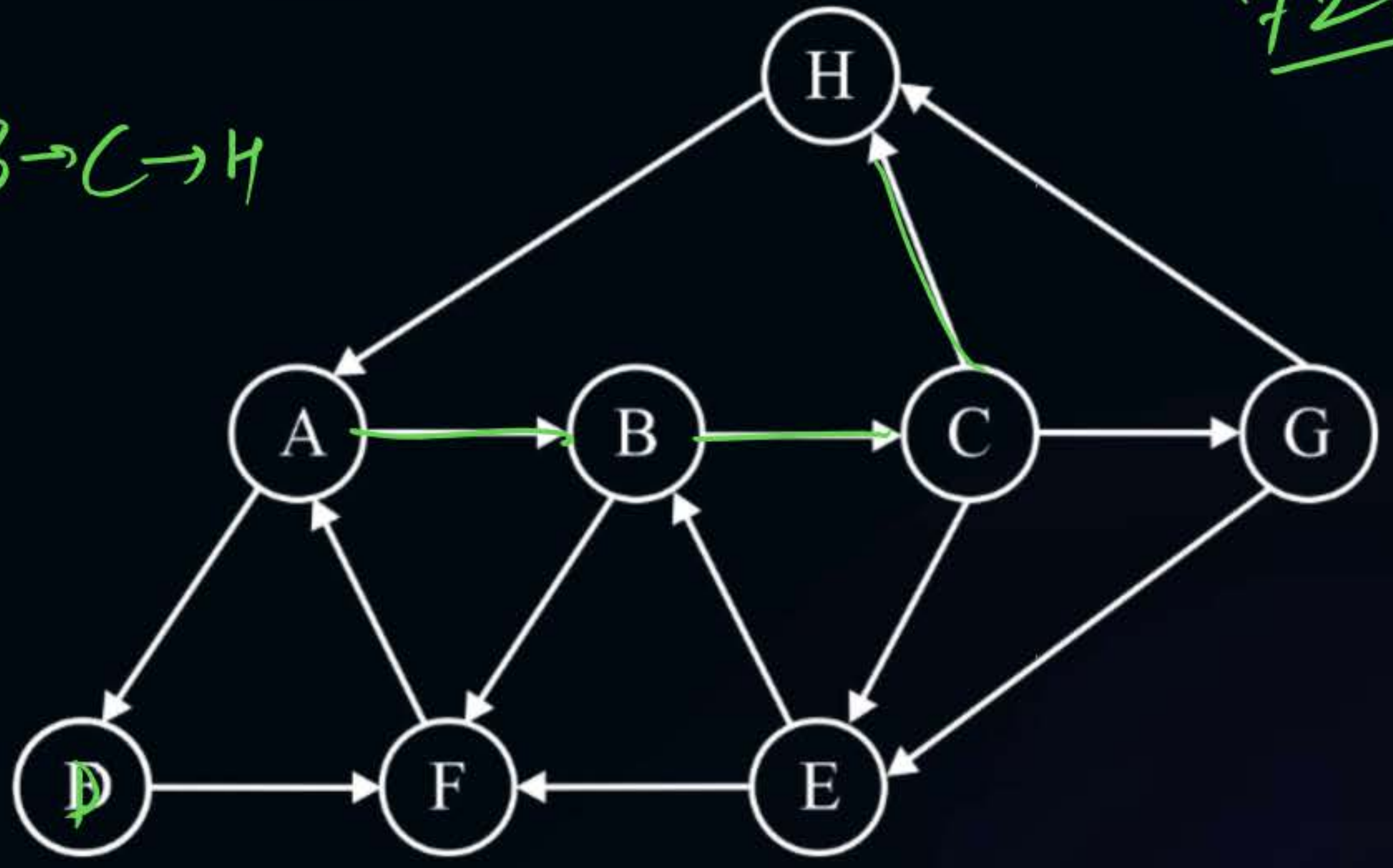
open

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

ABDCFEGH

Closed

goal



Topic : Uninformed Search

DFS $\Rightarrow$ tree

$A \rightarrow H$

$\Rightarrow$(Alphabetical order)

38%

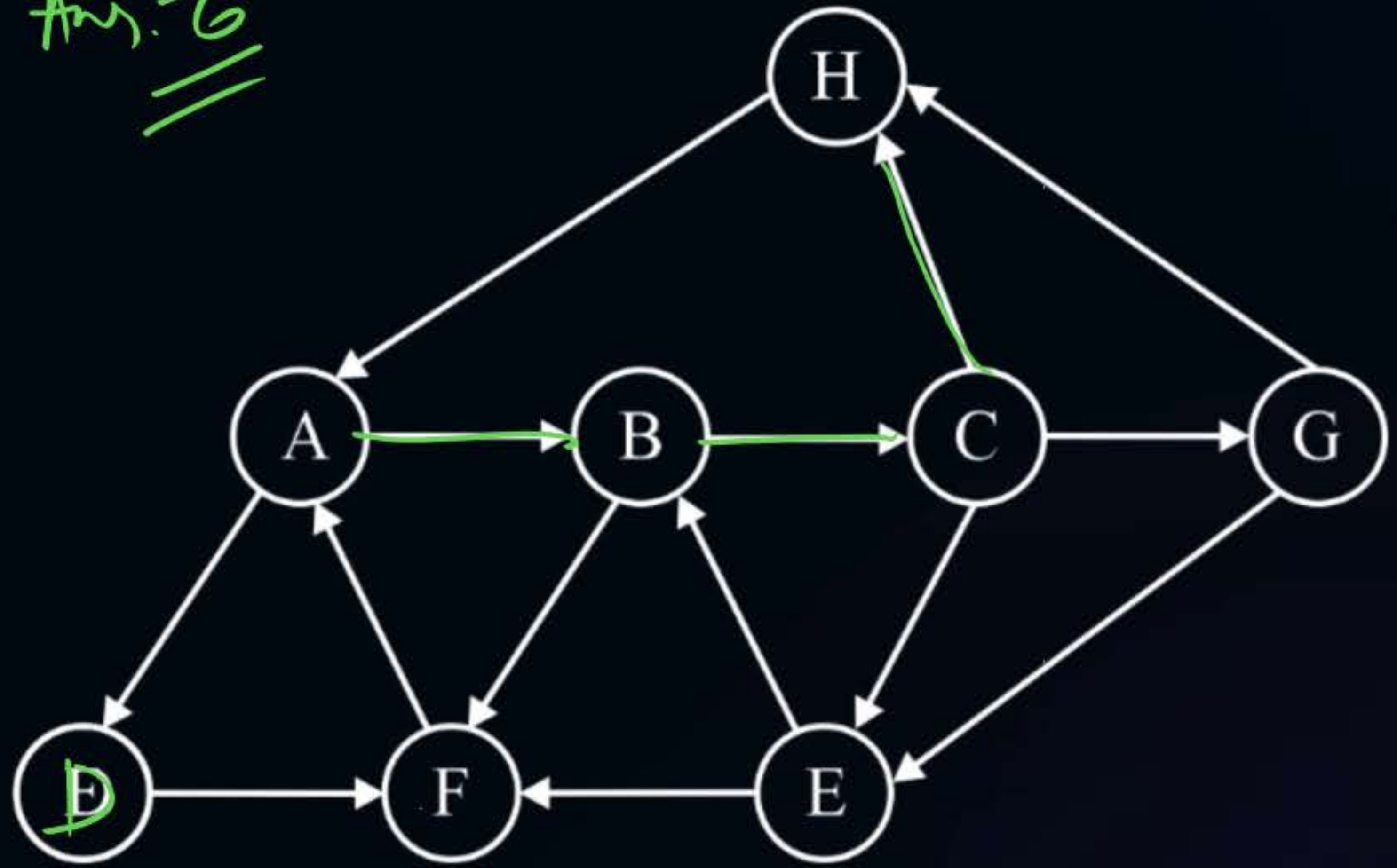
Ans:- G

2) DFS

$A \rightarrow H$

$\overset{1}{A} \rightarrow \overset{2}{B} \rightarrow \overset{3}{C} \rightarrow \overset{6}{G} \rightarrow \overset{7}{H}$

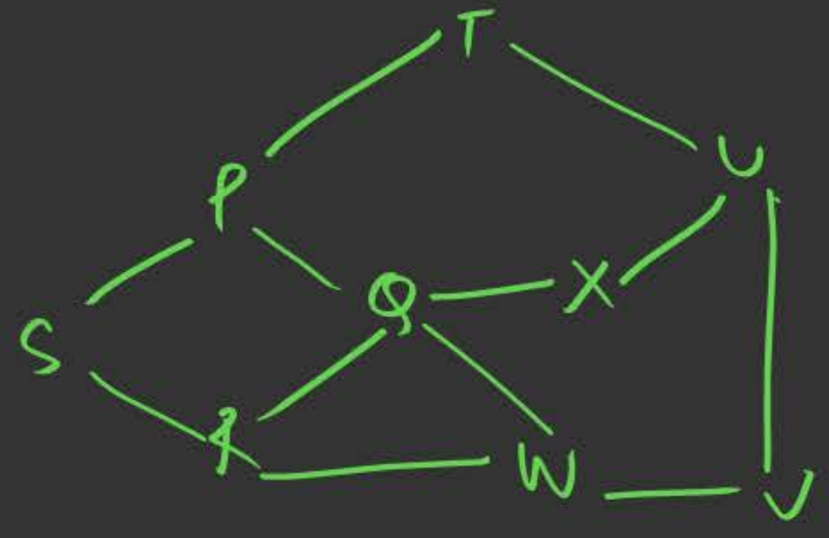
$\overset{3}{C} \rightarrow \overset{4}{E} \rightarrow \overset{5}{F}$



Start : S

Goal(s): U , X

1) BFS :



open

~~S~~ ~~P~~ ~~R~~ ~~Q~~ ~~T~~ ~~W~~ ~~X~~ U

S P R Q T W X

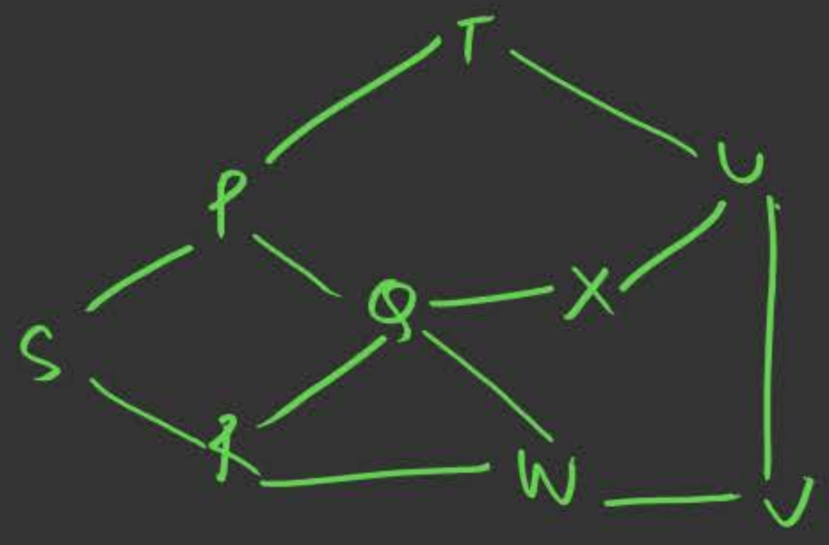
Closed

→ Stop

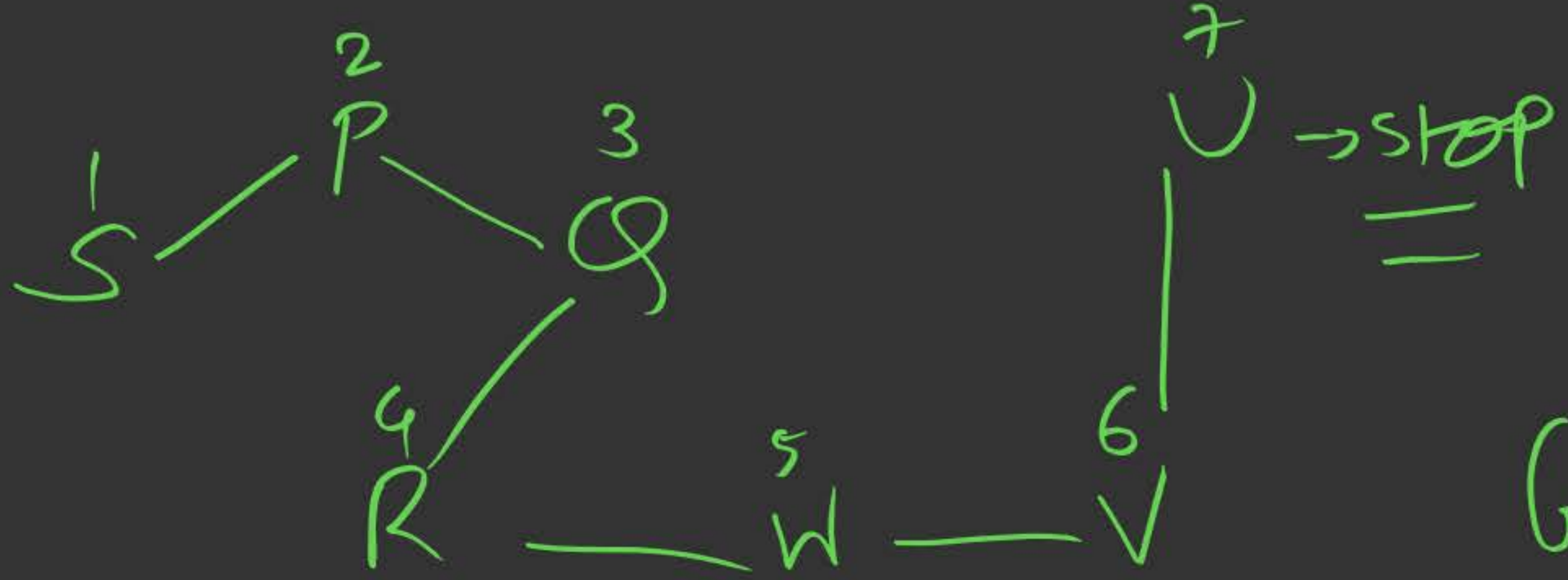
Goal : X

Start : S

Goal(s): U, X

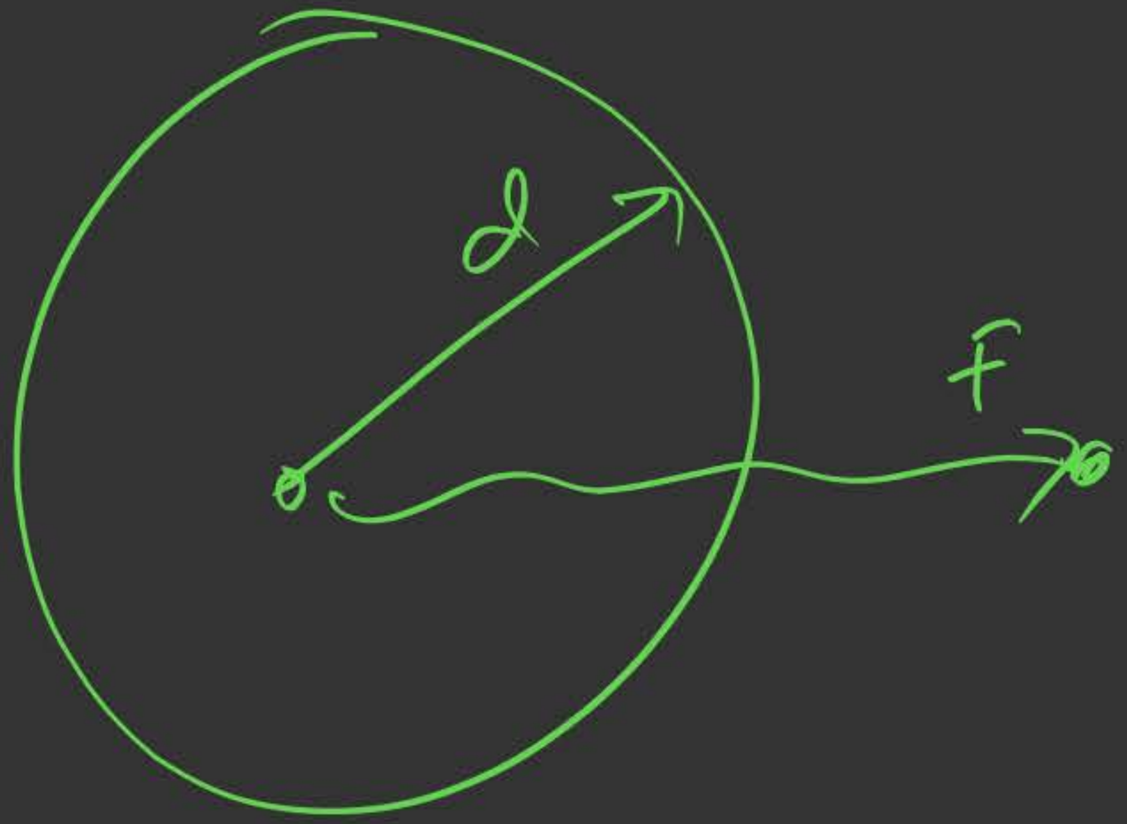


2) DFS



Goal: U

DLS

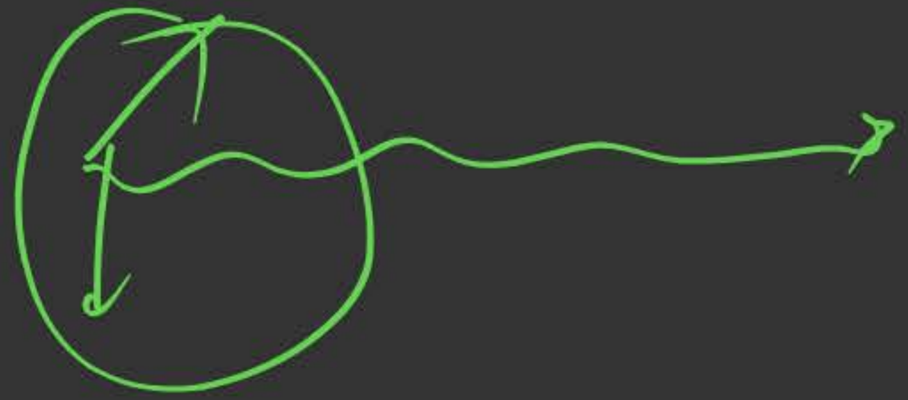


$f > d$

↳ not complete

if $d > f$

∞ search space



Depth Limited Search (DLS)

- Depth Limited Search is a modified version of DFS that imposes a limit on the depth of the search. This means that the algorithm will only explore nodes up to a certain depth, effectively preventing it from going down excessively deep paths that are unlikely to lead to the goal. By setting a maximum depth limit, DLS aims to improve efficiency and ensure more manageable search times.

Depth Limited Search (DLS)

- How Depth Limited Search Works
  - Initialization: Begin at the root node with a specified depth limit.
  - Exploration: Traverse the tree or graph, exploring each node's children.
  - Depth Check: If the current depth exceeds the set limit, stop exploring that path and backtrack.
  - Goal Check: If the goal node is found within the depth limit, the search is successful.
  - Backtracking: If the search reaches the depth limit or a leaf node without finding the goal, backtrack and explore other branches.

Topic : Uninformed Search

Depth Limited Search (DLS)

- Advantages of Depth Limited Search
- Depth limited search is better than DFS and requires less time and memory space.
- DFS assures that the solution will be found if it exists infinite time.
- There are applications of DLS in graph theory particularly similar to the DFS.
- To combat the disadvantages of DFS, we add a limit to the depth, and our search strategy performs recursively down the search tree.
- Disadvantages of Depth Limited Search
- The depth limit is compulsory for this algorithm to execute.
- The goal node may not exist in the depth limit set earlier, which will push the user to iterate further adding execution time.
- The goal node will not be found if it does not exist in the desired limit. $i > d$

Topic : Uninformed Search

b^d $b*d$. d b

depth
$L \rightarrow$ limit

Depth Limited Search (DLS)

- Performance Measures
- Completeness: The DLS is a complete algorithm in general except the case when the goal node is the shallowest node, and it is beyond the depth limit, i.e. l < d, and in this case, we never reach the goal node.
- Optimality: The DLS is a non-optimal algorithm since the depth that is chosen can be greater than d (l>d). Thus DLS is not optimal if l > d
- Time complexity is expressed as: It is similar to the DFS, i.e. ~~O(B1)~~ $O(B^L)$, where L is the set depth limit
- Space Complexity is expressed as: It is similar to DFS. O(BL), where L is specified depth limit

Iterative Deepening Search (IDS) or Iterative Deepening Depth First Search (IDDFS)

- There are two common ways to traverse a graph, BFS and DFS. Considering a Tree (or Graph) of huge height and width, both BFS and DFS are not very efficient due to following reasons.
- DFS first traverses nodes going through one adjacent of root, then next adjacent. The problem with this approach is, if there is a node close to root, but not in first few subtrees explored by DFS, then DFS reaches that node very late. Also, DFS may not find shortest path to a node (in terms of number of edges).
- BFS goes level by level, but requires more space.

Iterative Deepening Search (IDS) or Iterative Deepening Depth First Search (IDDFS)

√ imp

- Iterative Deepening Depth-First Search (IDDFS) combines the depth-first search's space efficiency with the breadth-first search's completeness. It repeatedly performs depth-limited searches with increasing depth limits until the goal is found.

DLS

Topic : Uninformed Search

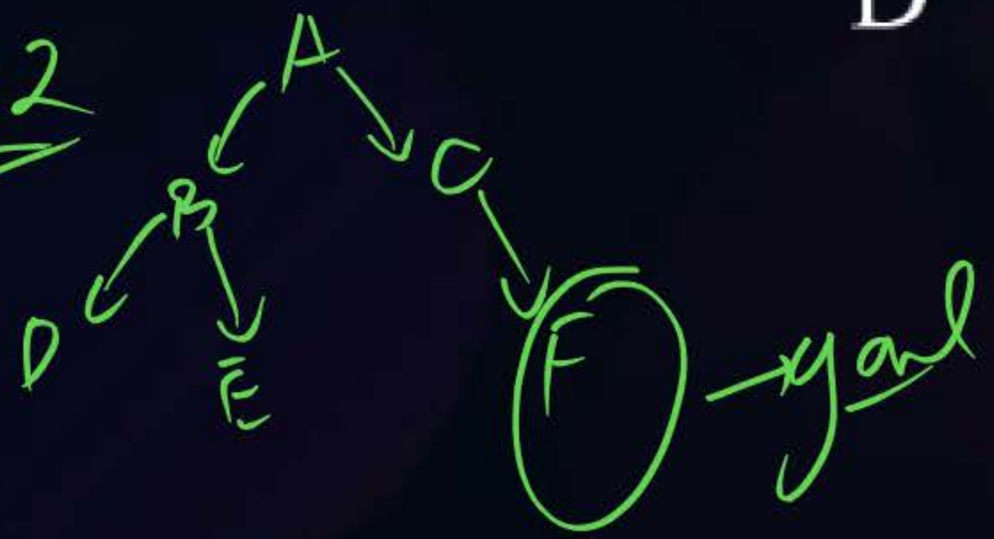
IDDFS

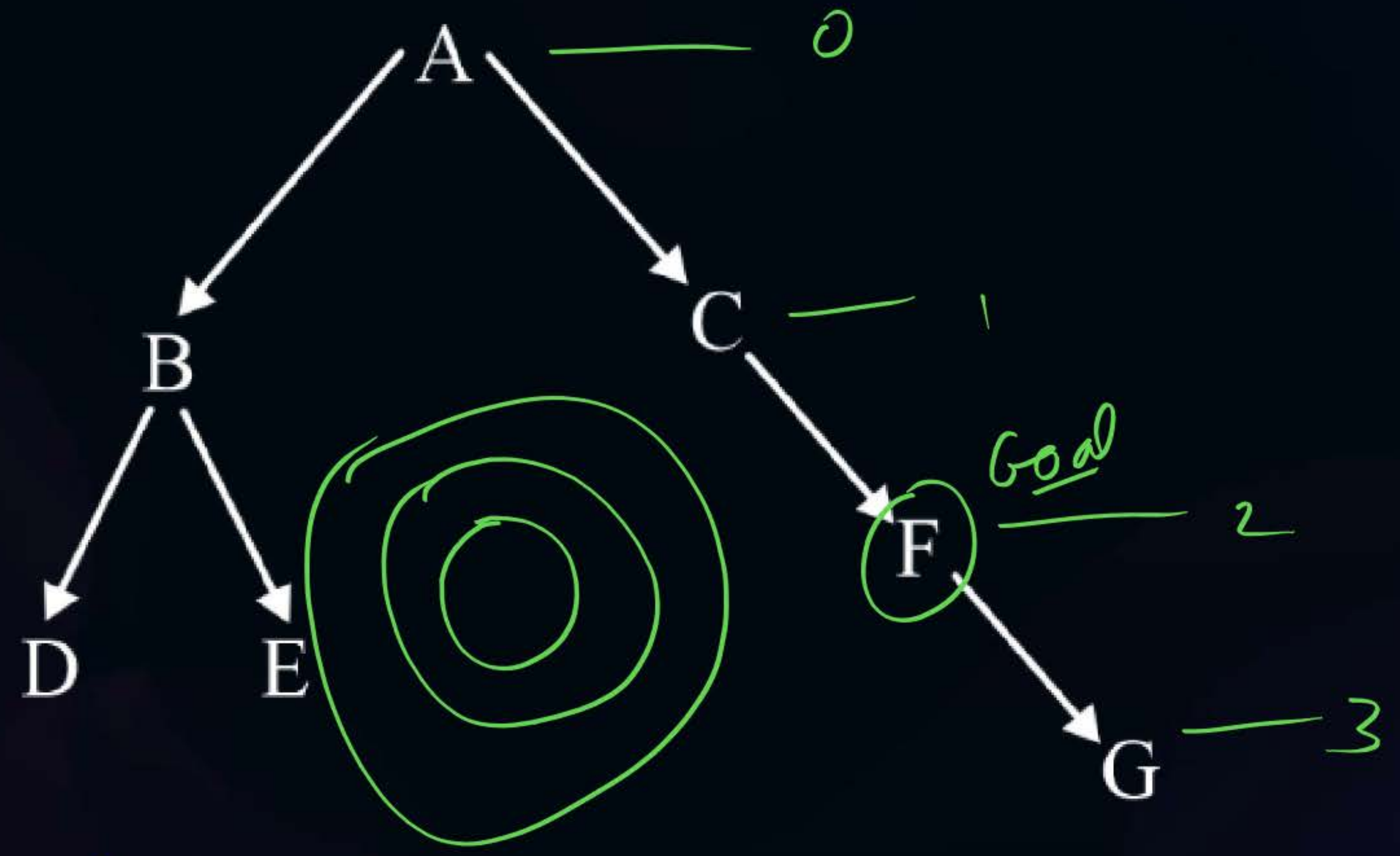
l = depth limit = 0

visited: A

$l = 1$

$l = 2$





Topic : Uninformed Search

Iterative Deepening Search (IDS) or Iterative Deepening Depth First Search (IDDFS)

Time Complexity:

- Let's suppose b is the branching factor and depth is d then the worst-case time complexity is $O(b^d)$.

Space Complexity:

- The space complexity of IDDFS will be $\cancel{O(b^d)}$.

$O(b * d)$

Iterative Deepening Search (IDS) or Iterative Deepening Depth First Search (IDDFS)

We will search for node N.

1) BFS

open

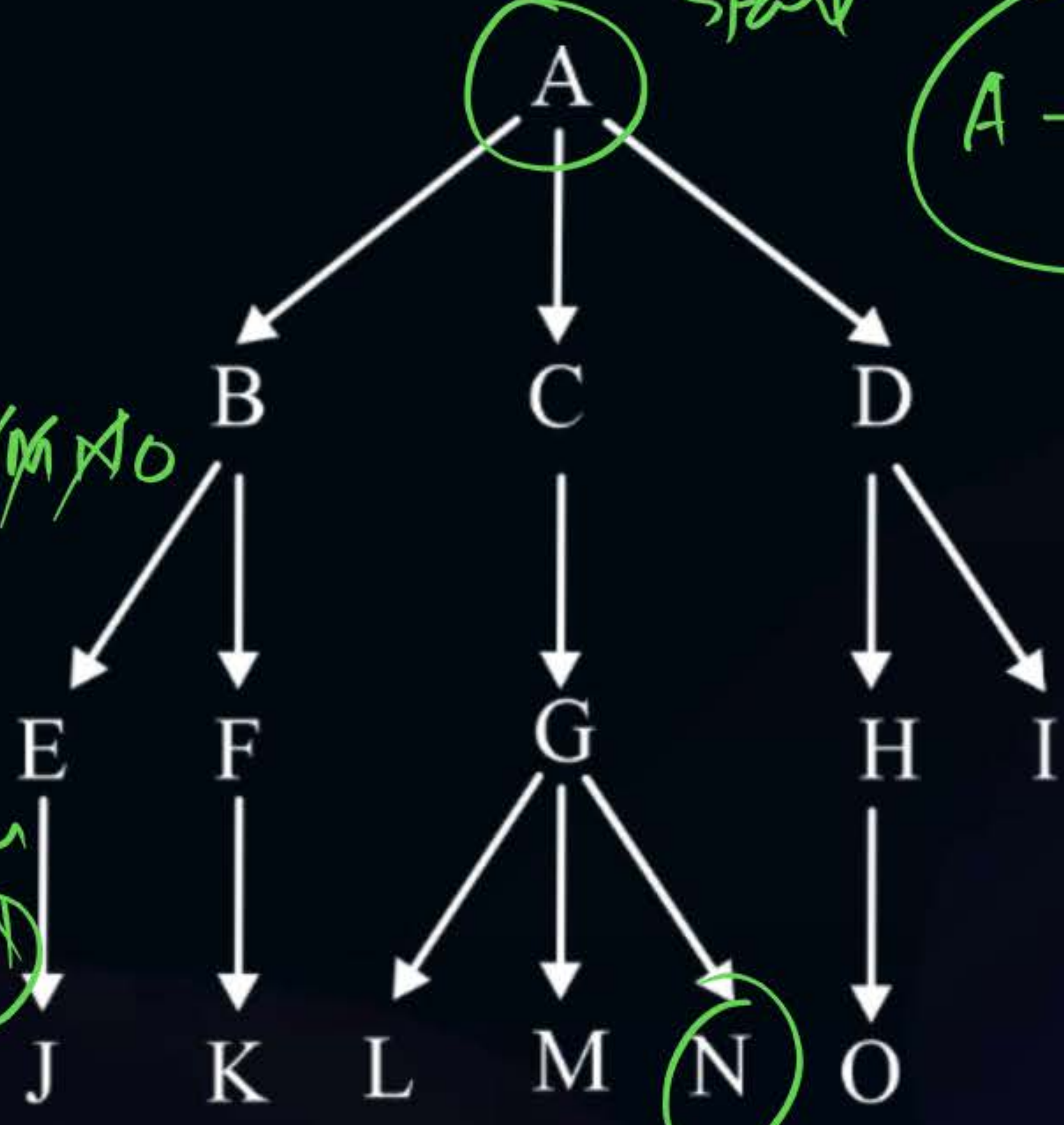
~~A B C D E F G H I J~~ K L M N O

closed

A B C D E F G H I J K L M N

start

A → C → G → N



Topic : Uninformed Search

Iterative Deepening Search (IDS) or Iterative Deepening Depth First Search (IDDFS)

We will search for node N.

2) DFS

Open

~~A D C B F E J K G N M L~~

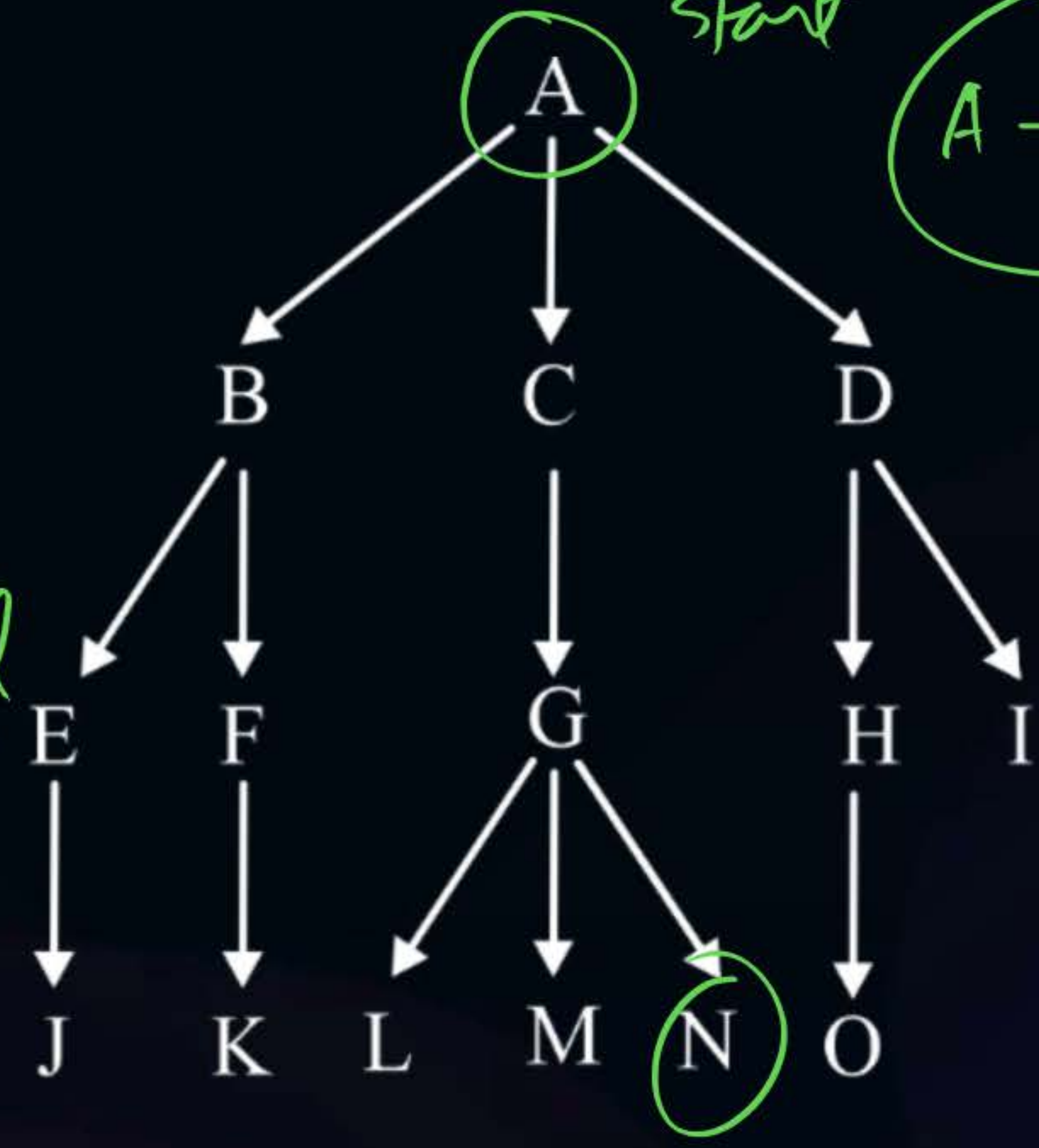
Closed

A B E J F K C G L M N

goal

start

A → C → G → N



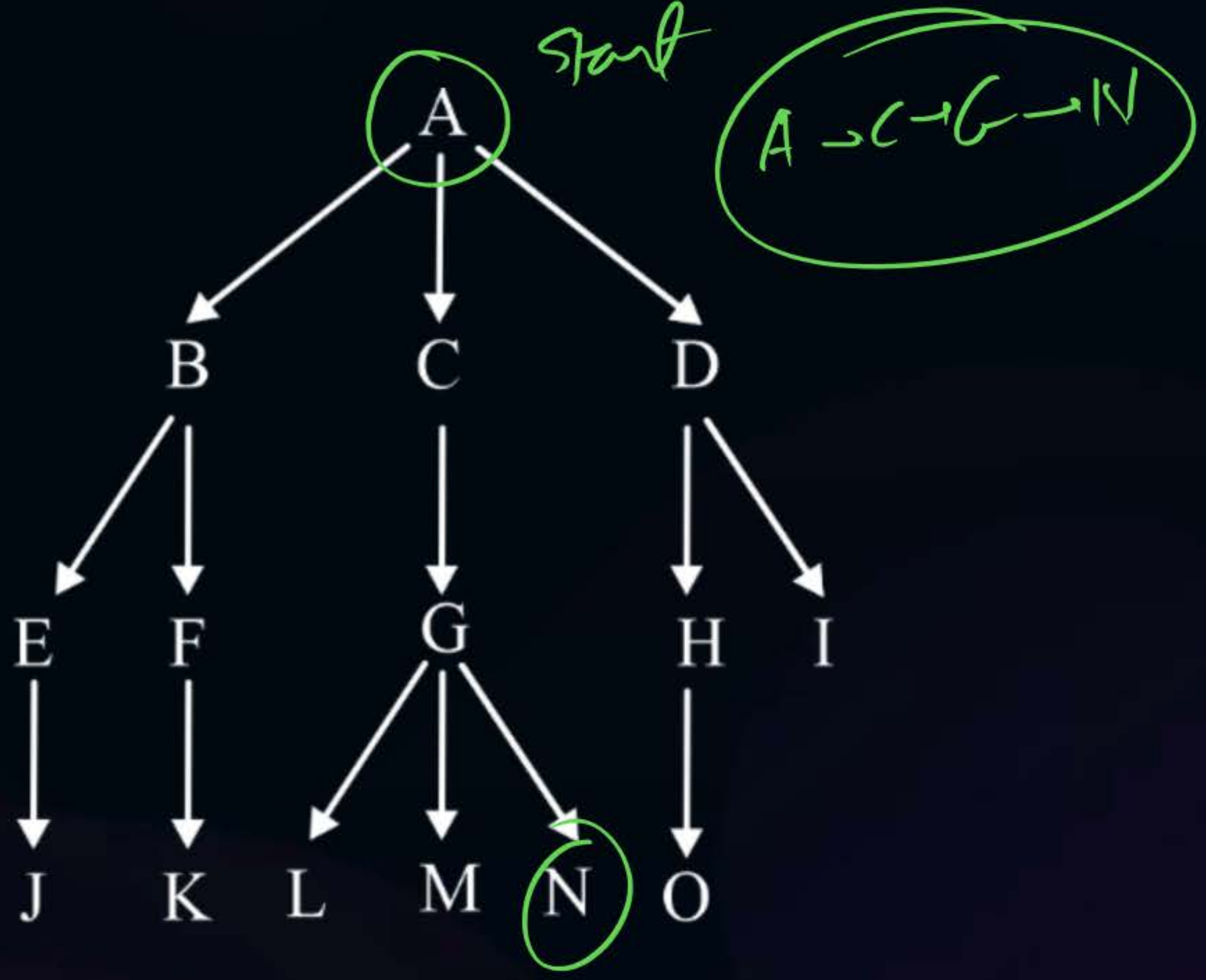
Iterative Deepening Search (IDS) or Iterative Deepening Depth First Search (IDDFS)

We will search for node N.

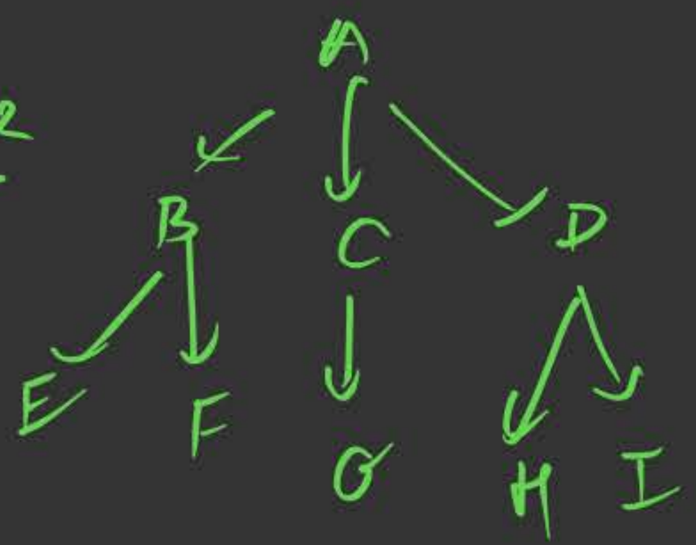
3) IDDFS

d = 0   A

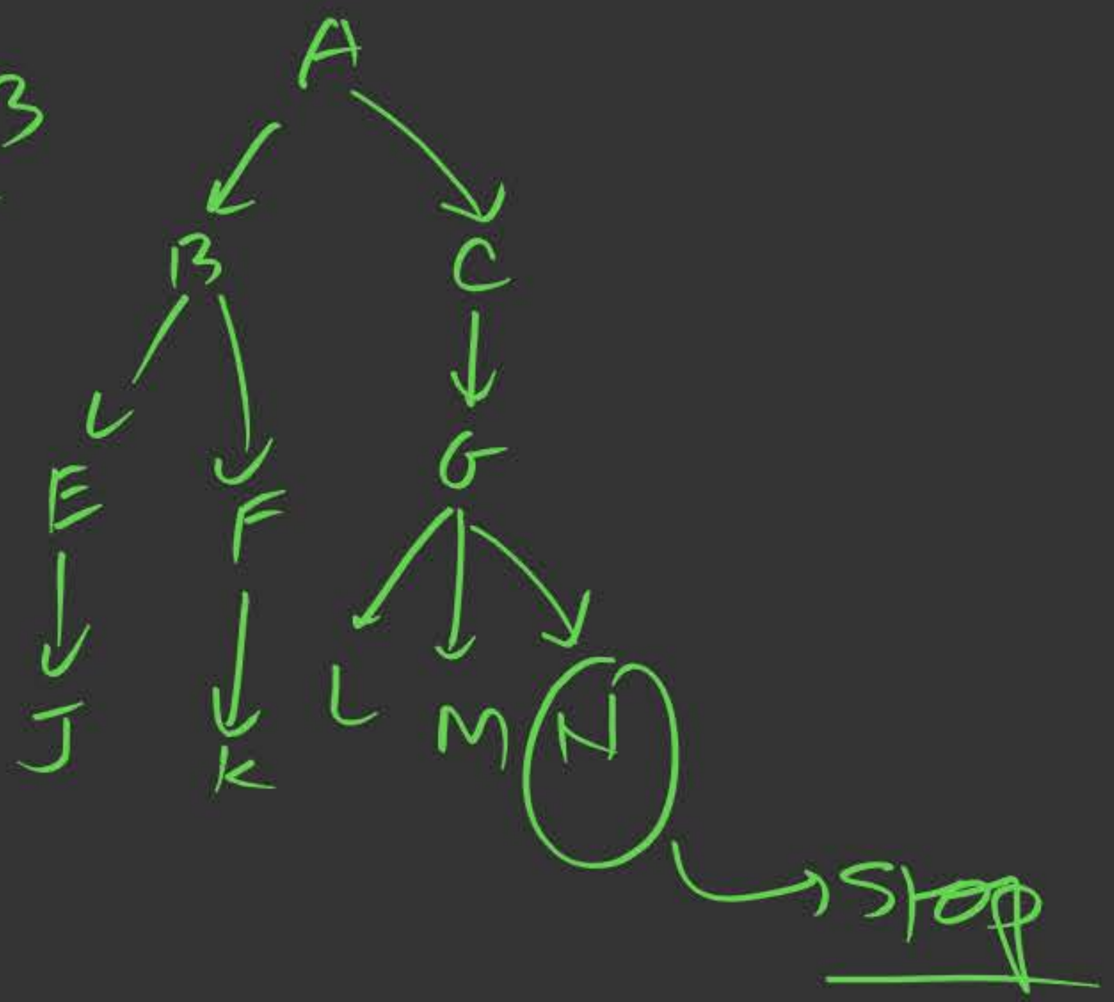
d = 1   A → B, C, D



d = 2



d = 3



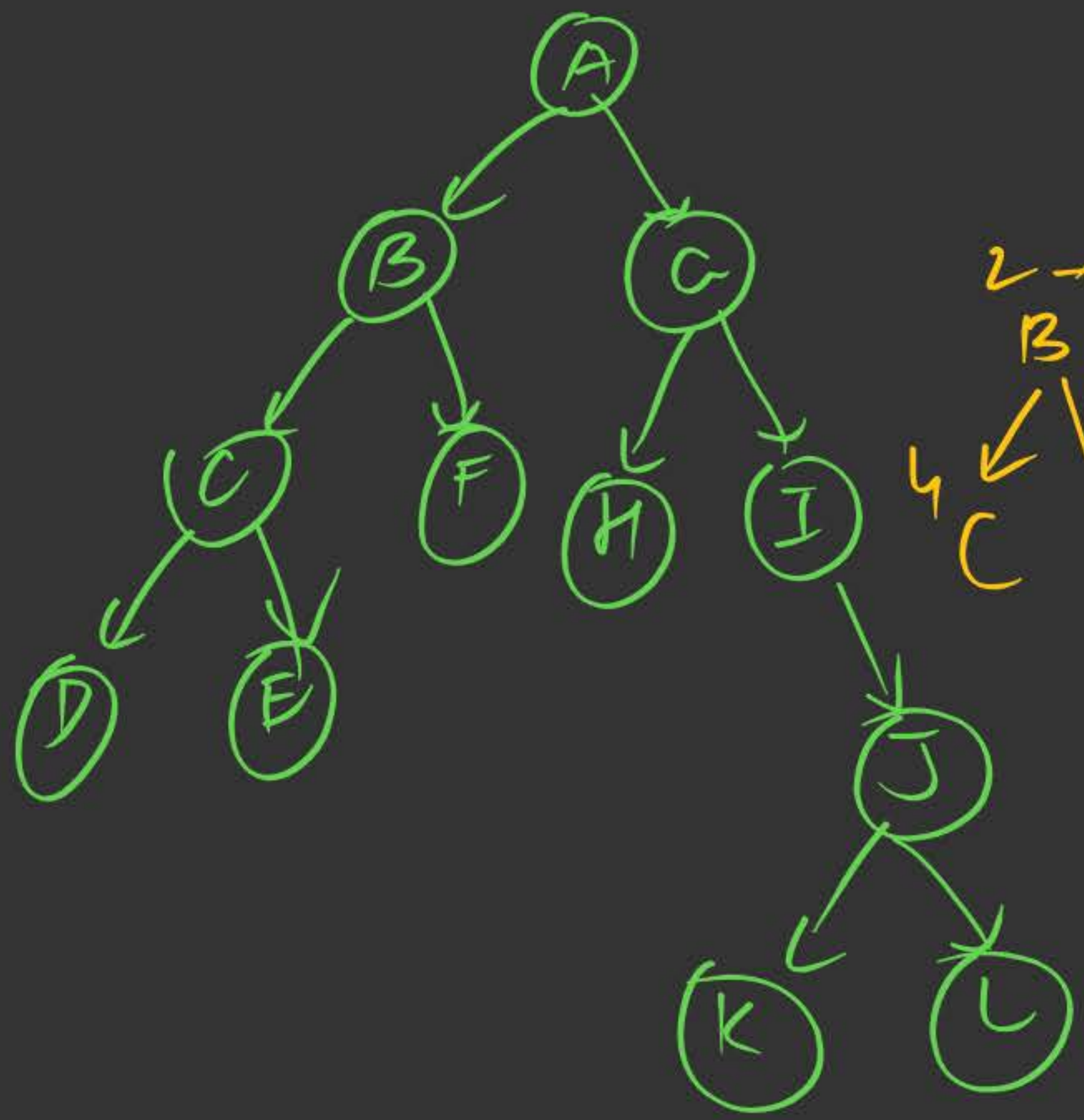
Iterative Deepening Search (IDS) or Iterative Deepening Depth First Search (IDDFS)

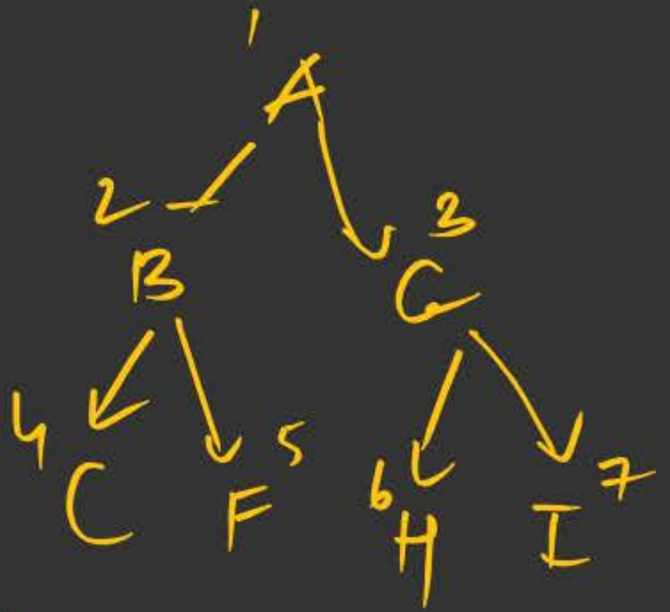
- Space Complexity
- Each iteration of depth-limited DFS uses space proportional to the depth limit.
- IDDFS does not store all nodes at a given depth level simultaneously; it only stores the path from the root to the current node.

Start: A

Goal: I.

1) BFS

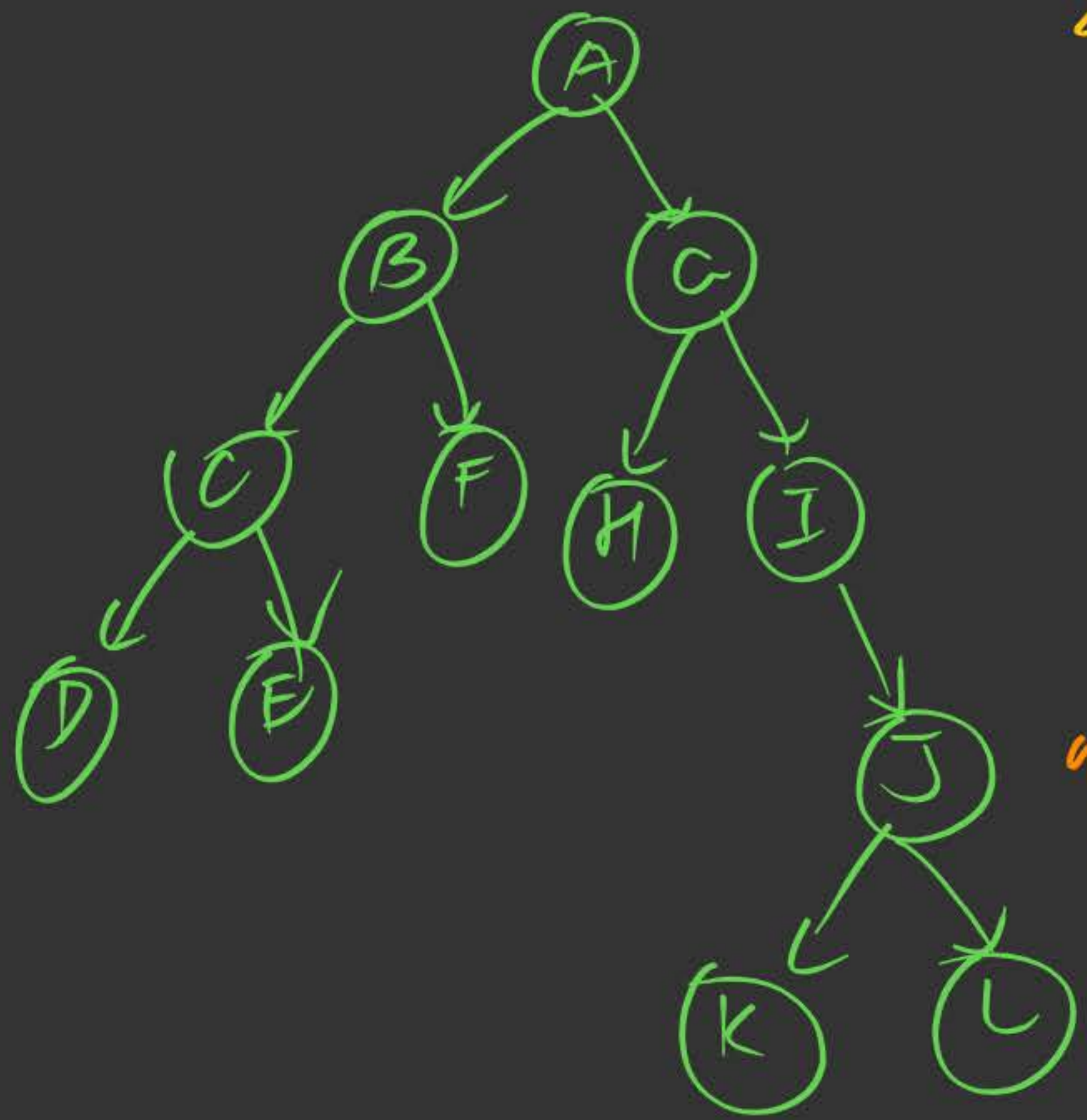


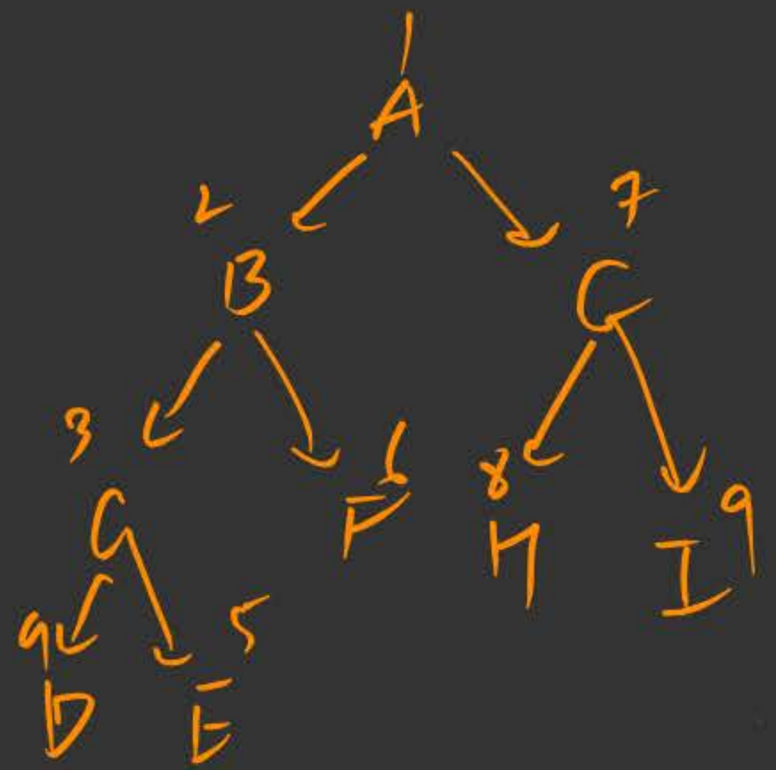


73%

Start: A
Goal: I.

2) DFS



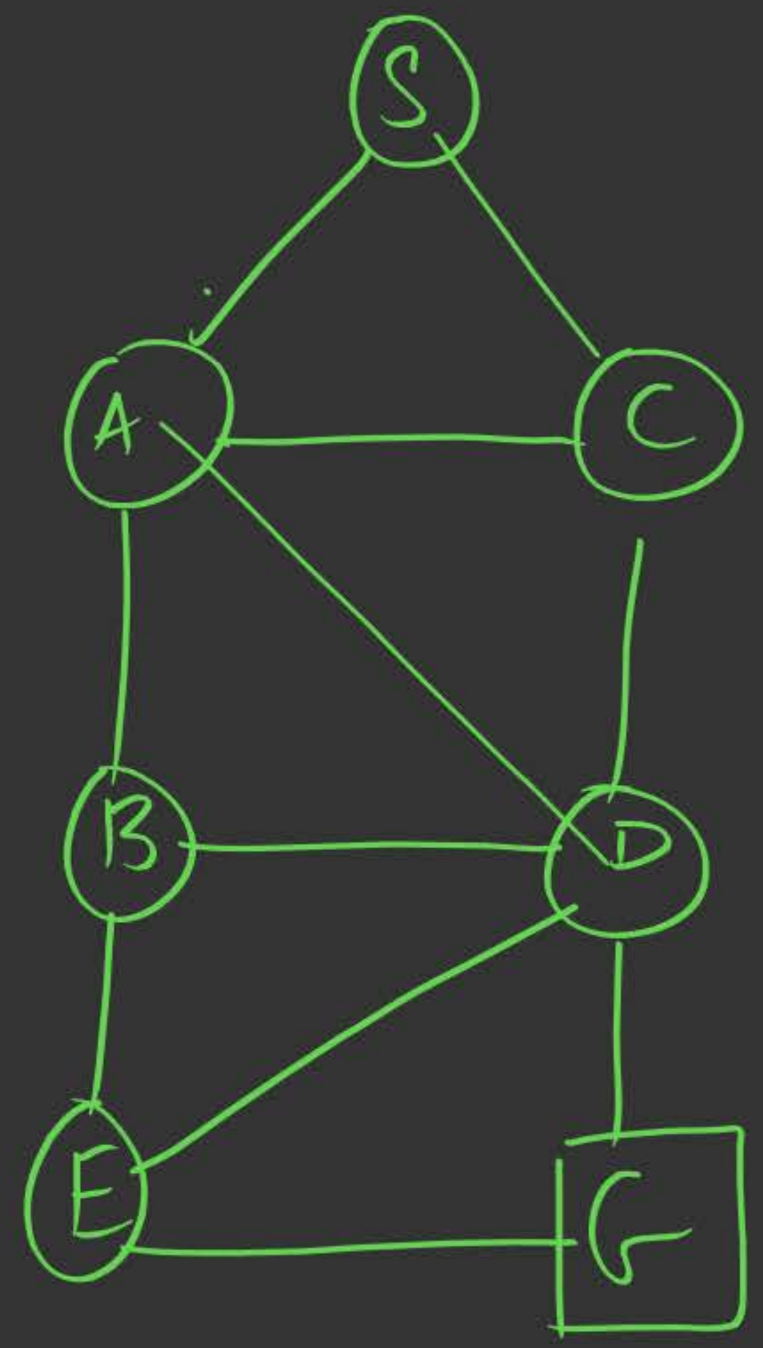


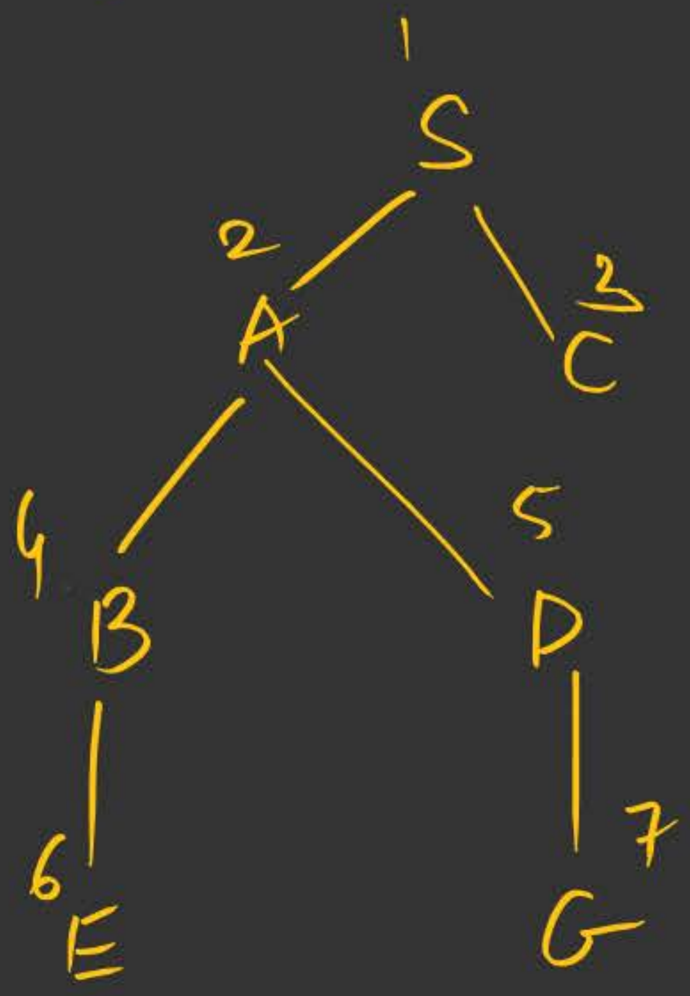
85%

(Q)

$S \longrightarrow G$ alphabetical

i) Draw BFS (ST)



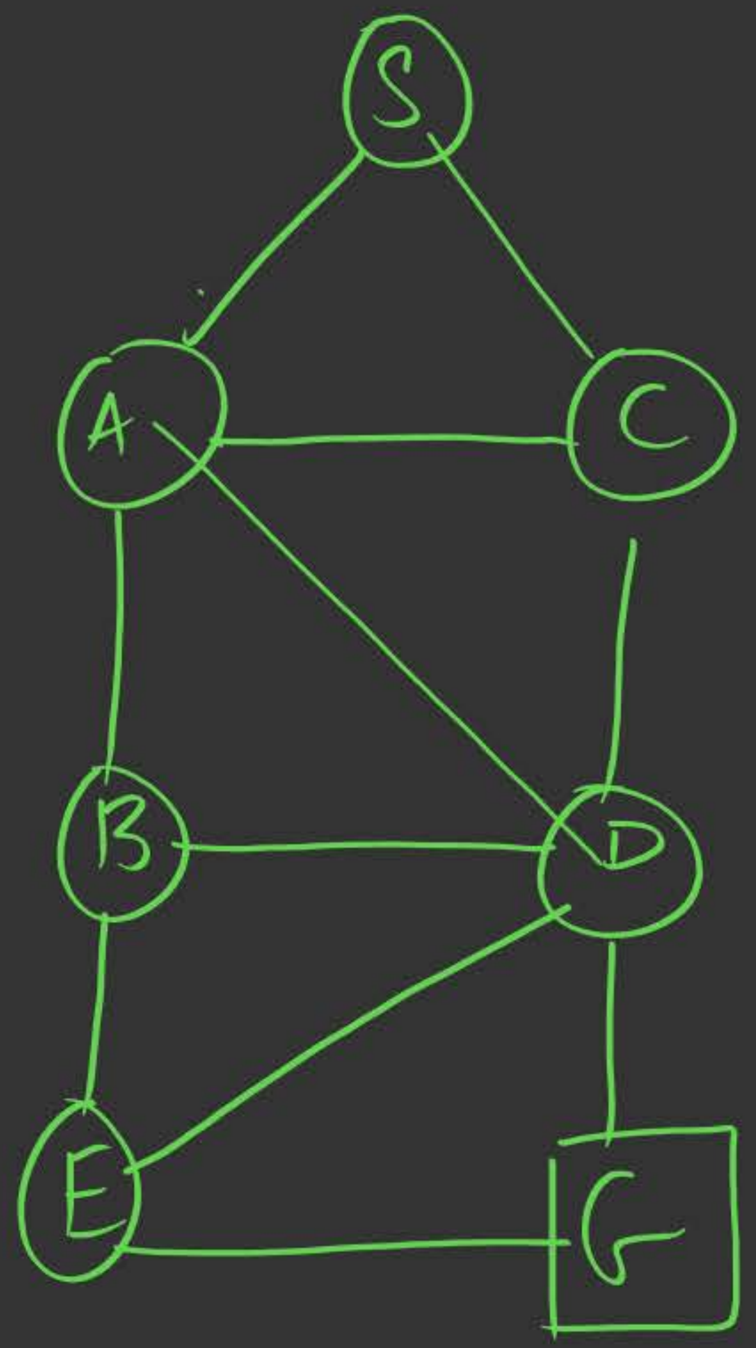


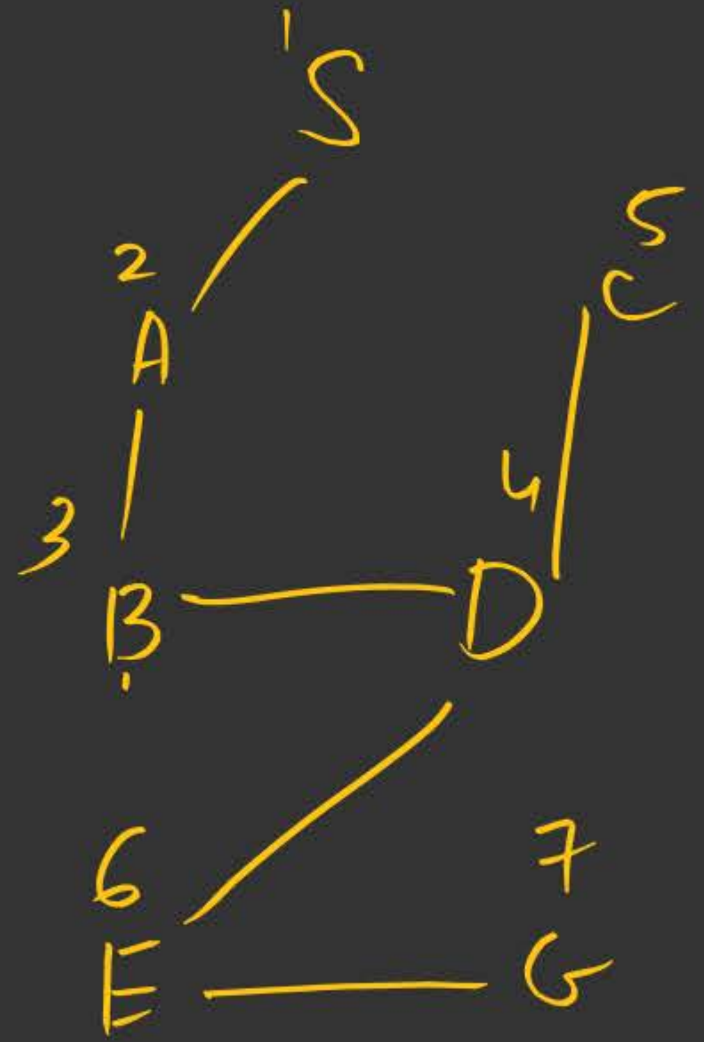
Path

(Q)

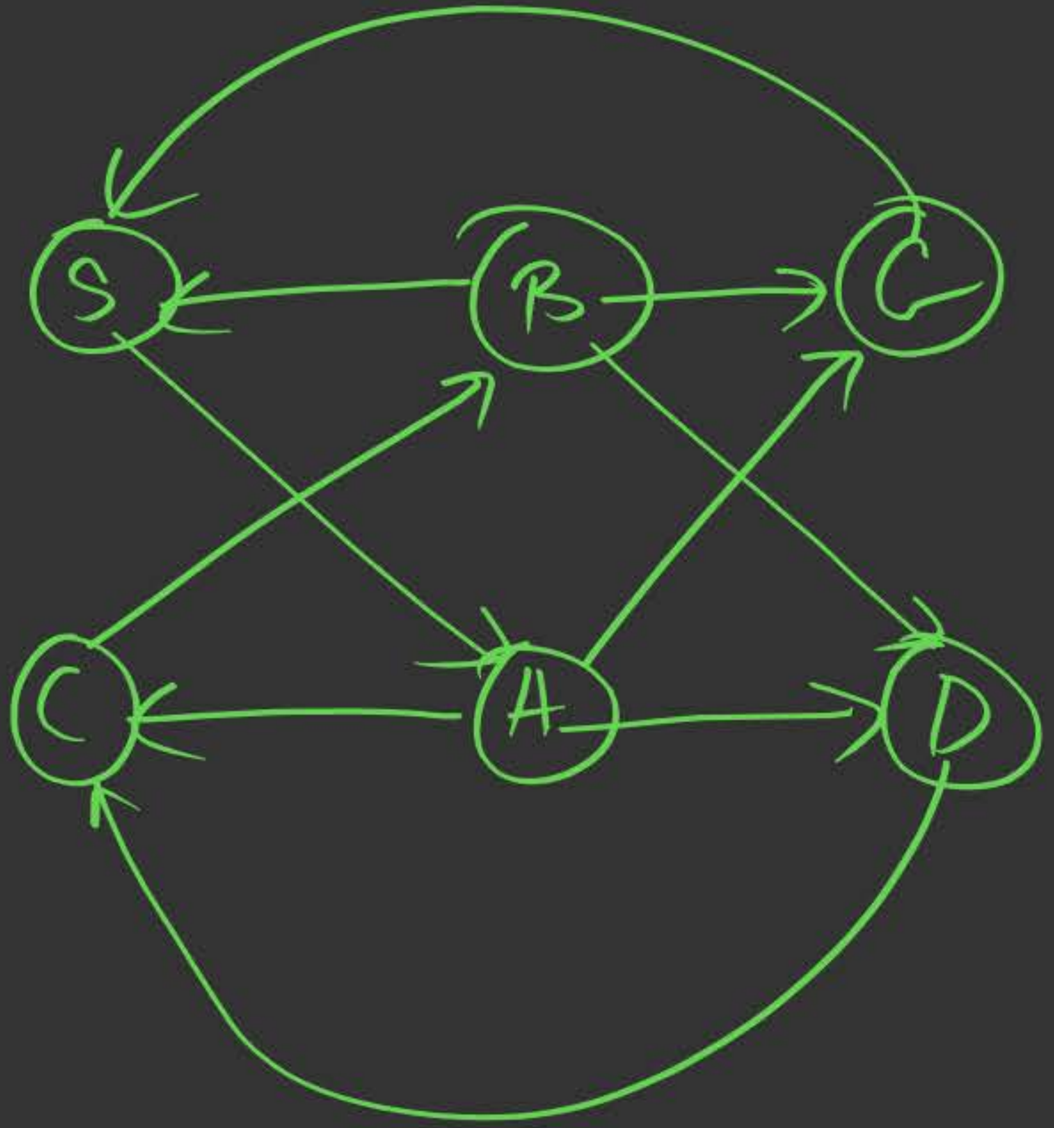
$S \longrightarrow G$ alphabetical

1) Draw DFS (ST)



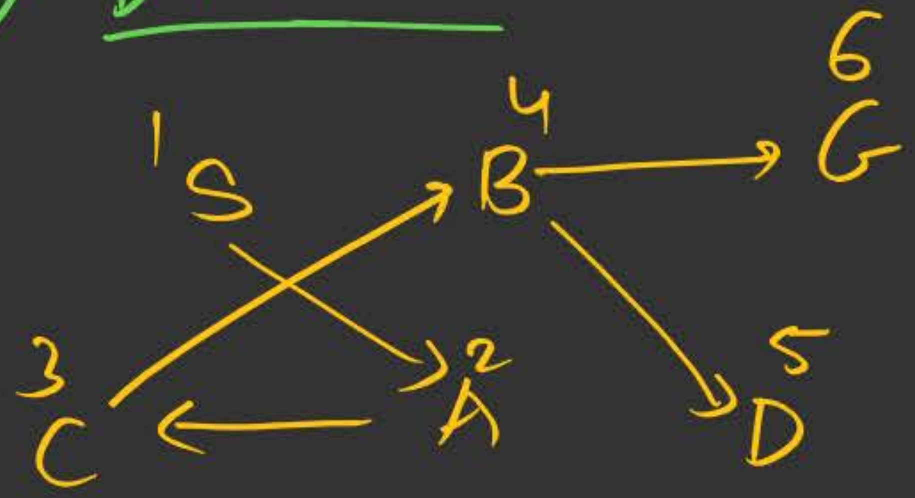


(Q)

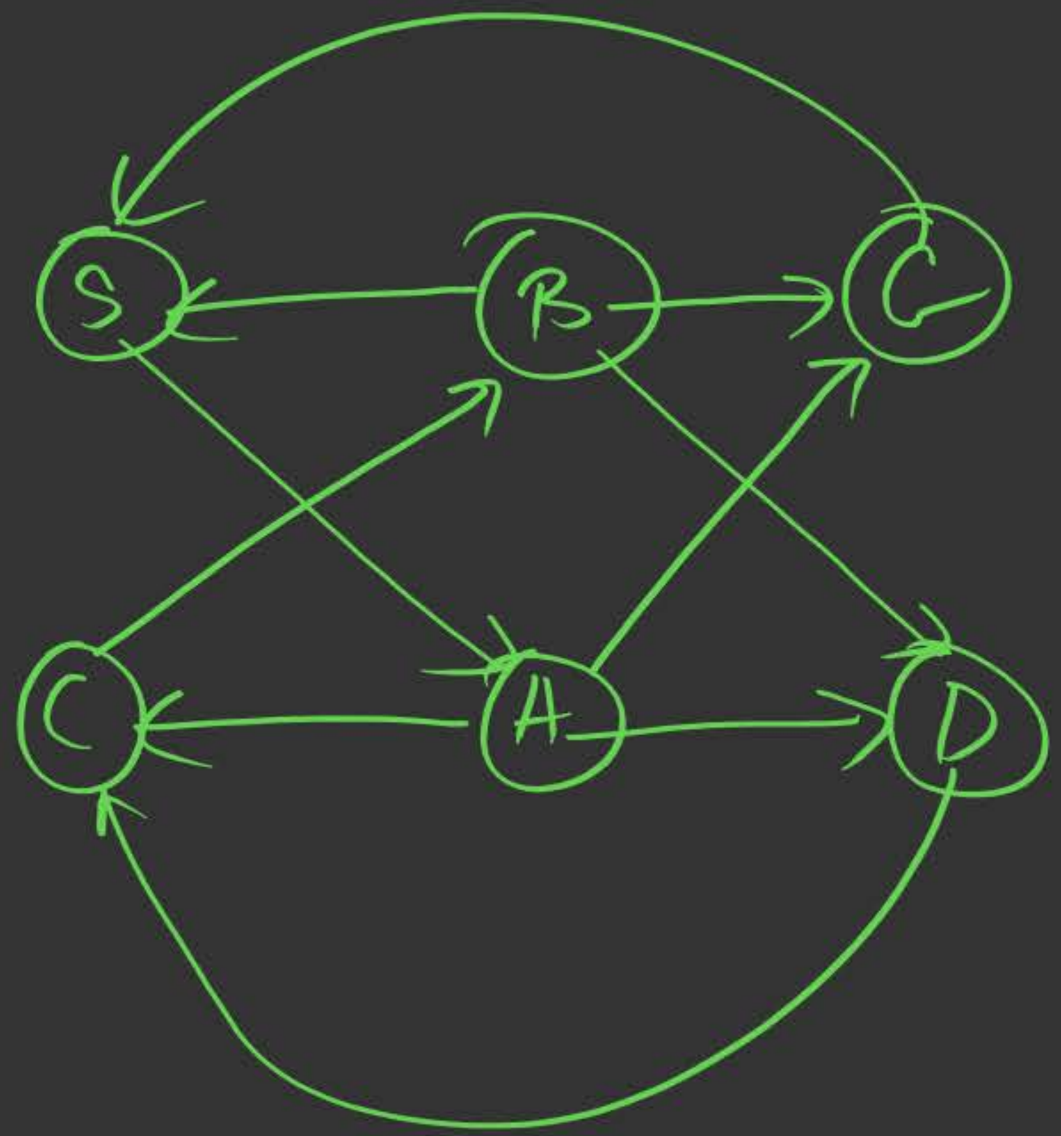


$S \to G$

① DFS ST

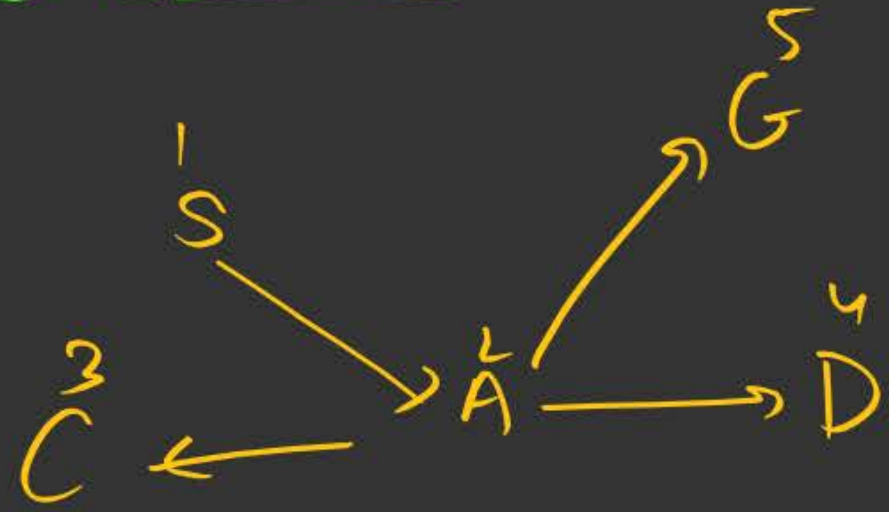


(Q)



$S \to G$

② BFS ST



Uniform Cost Search

Uniform Cost Search (UCS) is a search algorithm used to find the least-cost path from a start node to a goal node in a graph. UCS is a variant of Dijkstra's algorithm and is used in situations where the path costs vary, making it more suitable for weighted graphs.

BFS shortest → less edges

UCS shortest → path cost

Uniform Cost Search

Key Concepts

- Priority Queue UCS uses a priority queue to explore the least-cost path first. Nodes are expanded in order of their cumulative cost from the start node.
- Cost: Each edge in the graph has a cost associated with it. UCS aims to find the path with the minimum cumulative cost.
- Expansion: The algorithm expands the node with the smallest cumulative cost, ensuring that the cheapest path is always chosen.

Topic : Uninformed Search

SSSP Algo

$S \longrightarrow G$

Start = A

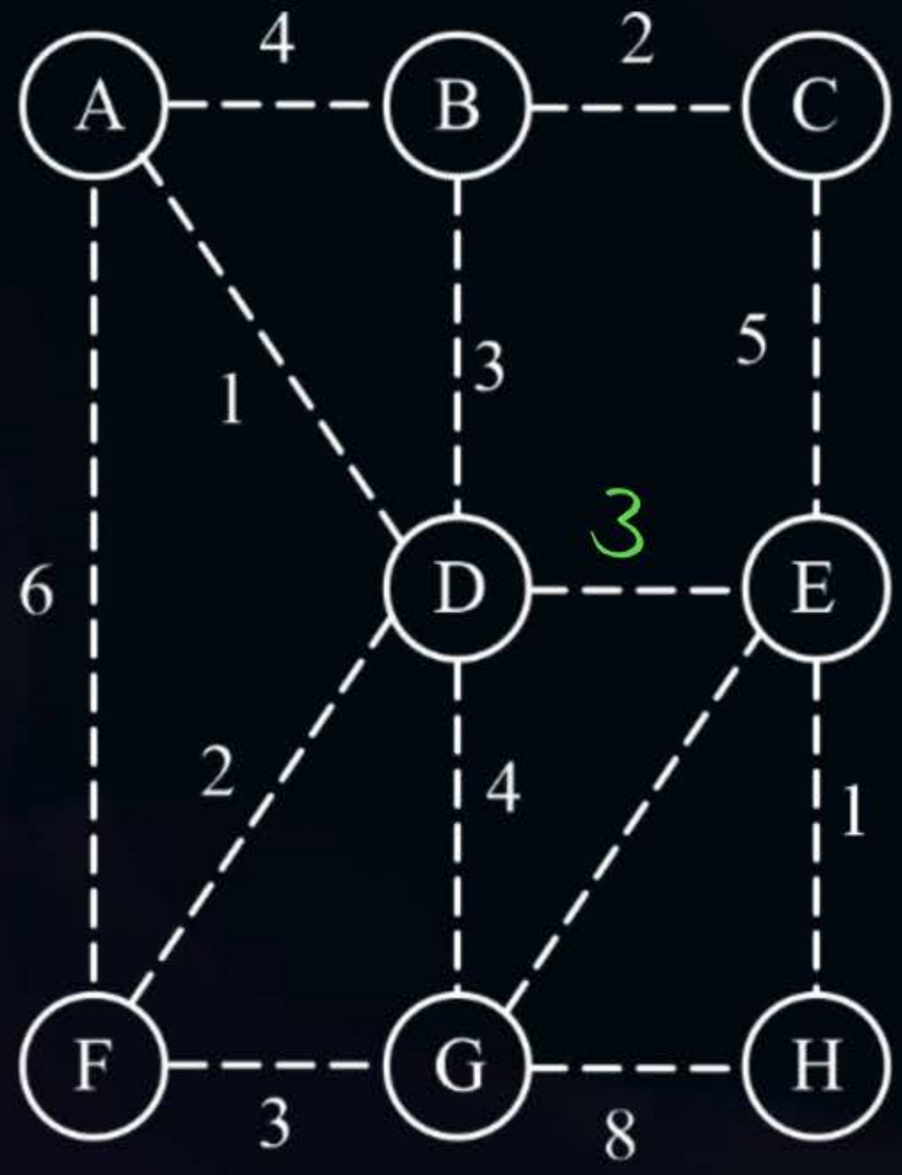
Goal = H

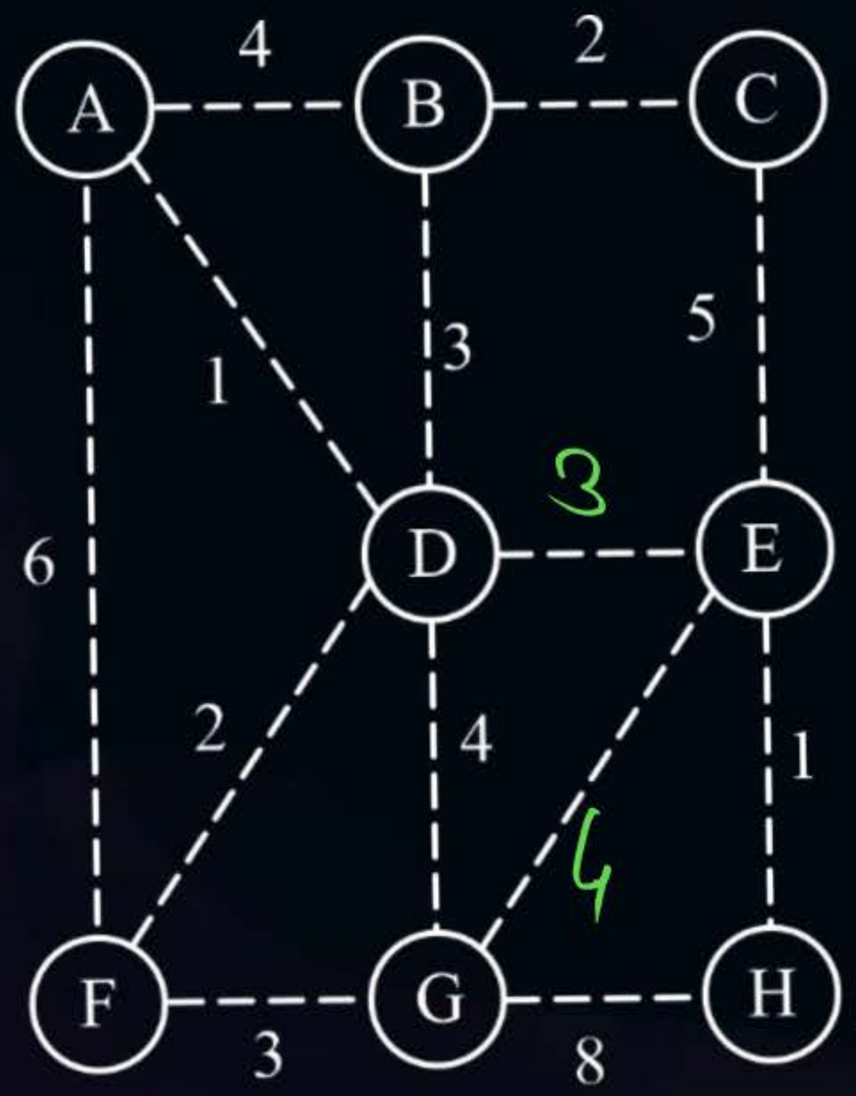
Strategy: UCS.

Ans:-

$A \longrightarrow H : 5$

path: $A \to D \to E \to H$





Closed/ Visited

Start

{ }

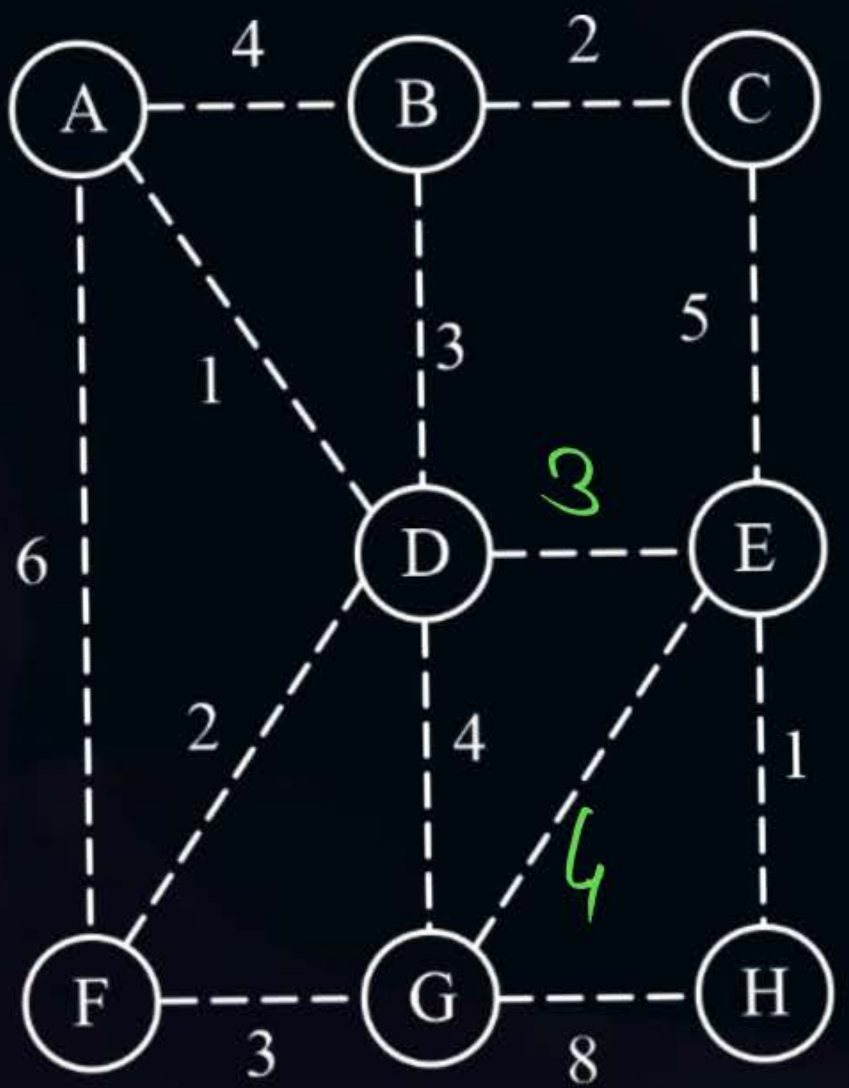
→ cond

1) Dijkstra's SSSP

Closed/Visited	A	B	C	D	E	F	G	H
(A)	0	4	X	1	X	6	X	X
(A,D)	0	4	X	1	4	3	5	X
(A,D,F)	0	4	X	1	4	3	5	X
(A,D,F,B)	0	4	6	1	4	3	5	X
(A,D,F,B,E)	0	4	6	1	4	3	5	5
(A,D,F,B,E,G)	0	4	6	1	4	3	5	5
(A,D,F,B,E,G,H)	0	4	6	1	4	3	5	5

→ Goal

0



2) UCS

Open	Closed: A	D	F	B	E	G
x ~~A~~						
x B	4	4	4	4	4	4
x ~~D~~	1	1	1	1	1	1
x F	6	3	3	3	3	3
x E	x	4	4	4	4	4
G	x	5	5	5	5	5
C	x	x	x	6	6	6
x H	x	x	x	x	5	5 → goal

THANK - YOU