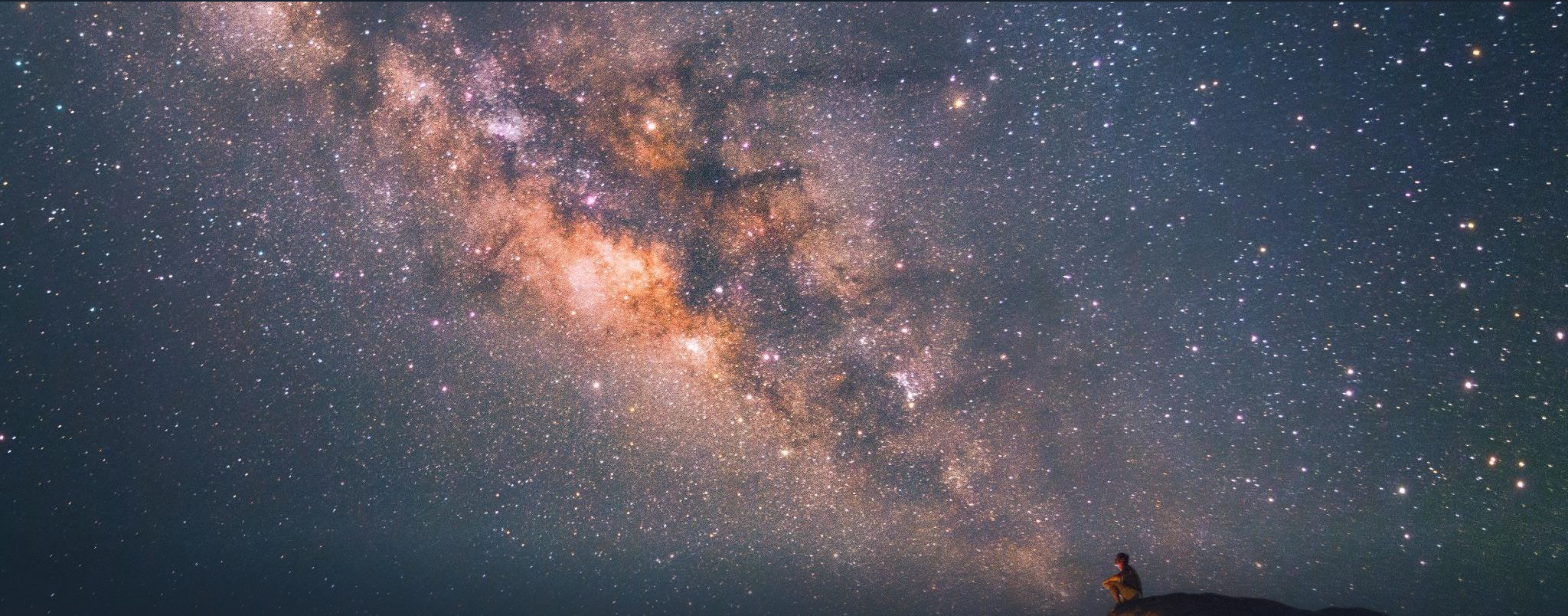
Module 1 Chapter 2

Artificial Intelligence

VBDS1402

Uninformed Search



Types of Search

Uninformed Search

Informed Search

Uninformed Search (Blind Search)

The term means that the strategies have no additional information about states beyond that provided in the problem definition.

All they can do is generate successors and distinguish a goal state from a non-goal state.

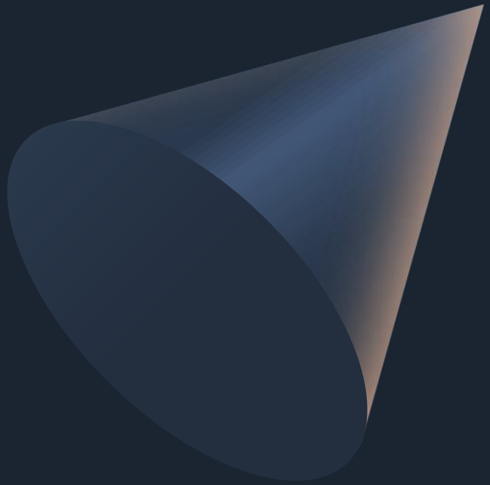
Types of Uninformed Search

- Breadth First
- Depth – first
- Uniform Cost
- Depth Limited
- Iterative deepening DFS
- Bidirectional

In this session you will learn:

Uninformed Search Methods

- Breadth First Search



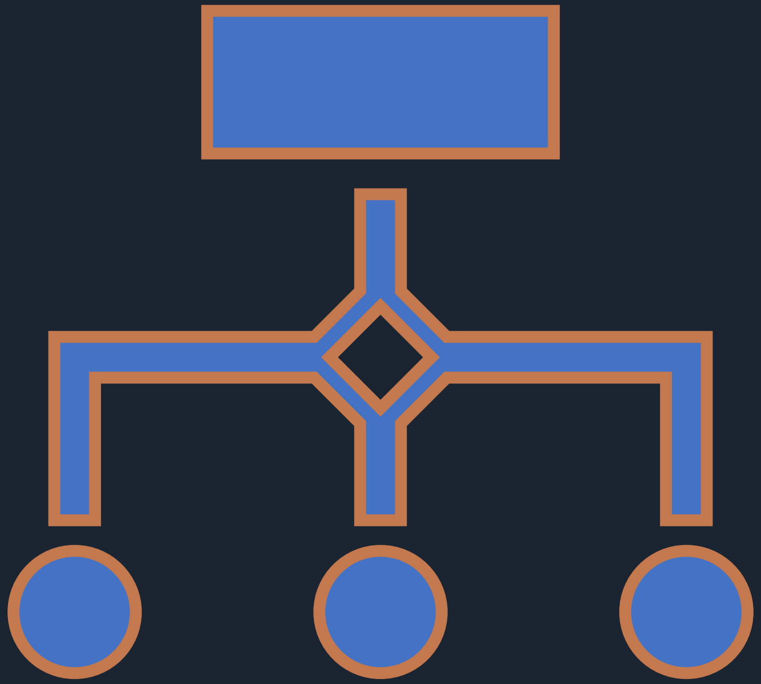
Breadth First Search (Uninformed/Blind Search)

- Breadth-first search is a simple strategy in which the root node is expanded first, then all the successors of the root node are expanded next, then their successors, and so on.

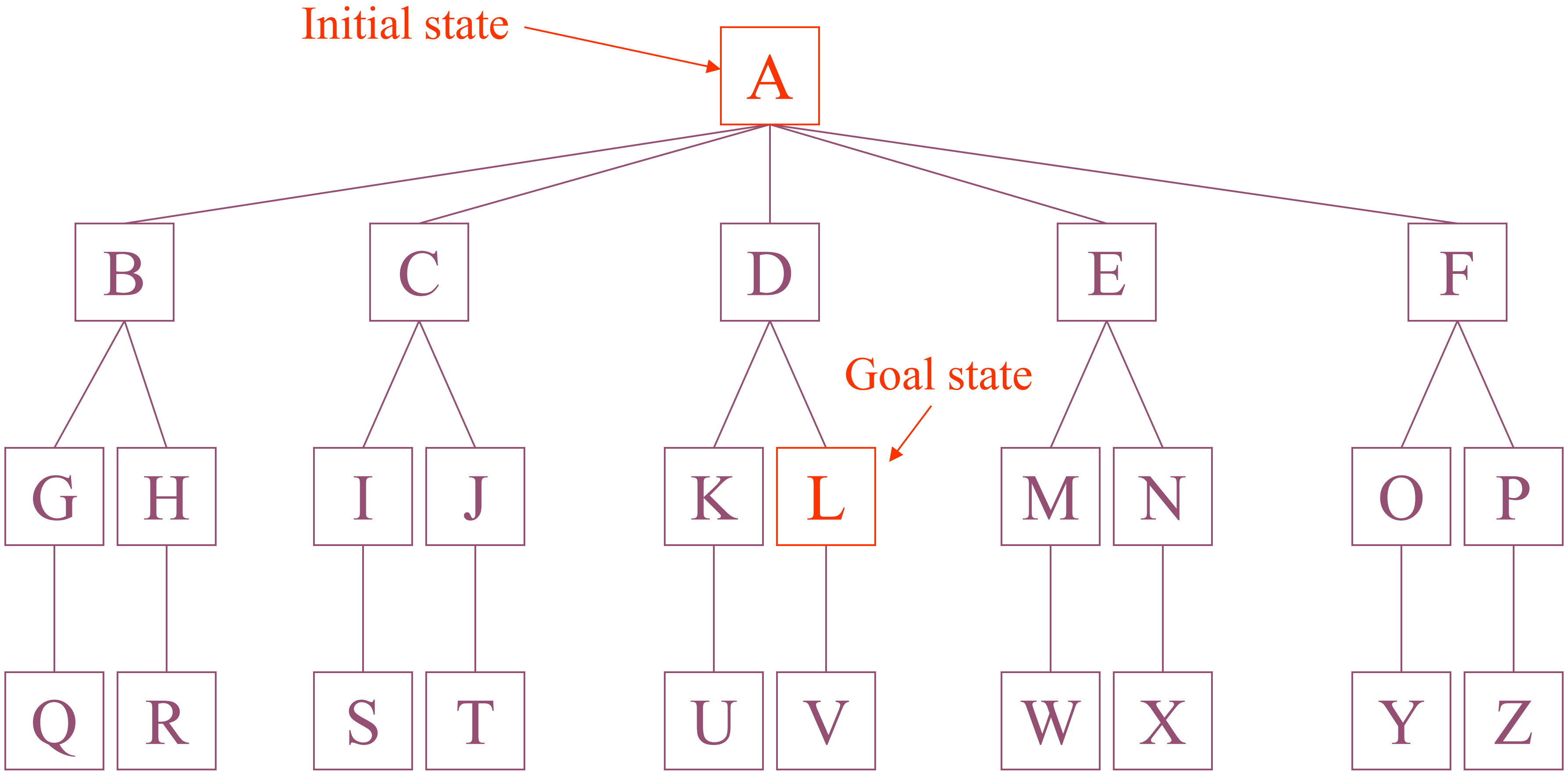
- In general, all the nodes are expanded at a given depth in the search tree before any nodes at the next level are expanded.

- This is achieved very simply by using a FIFO queue for the frontier.

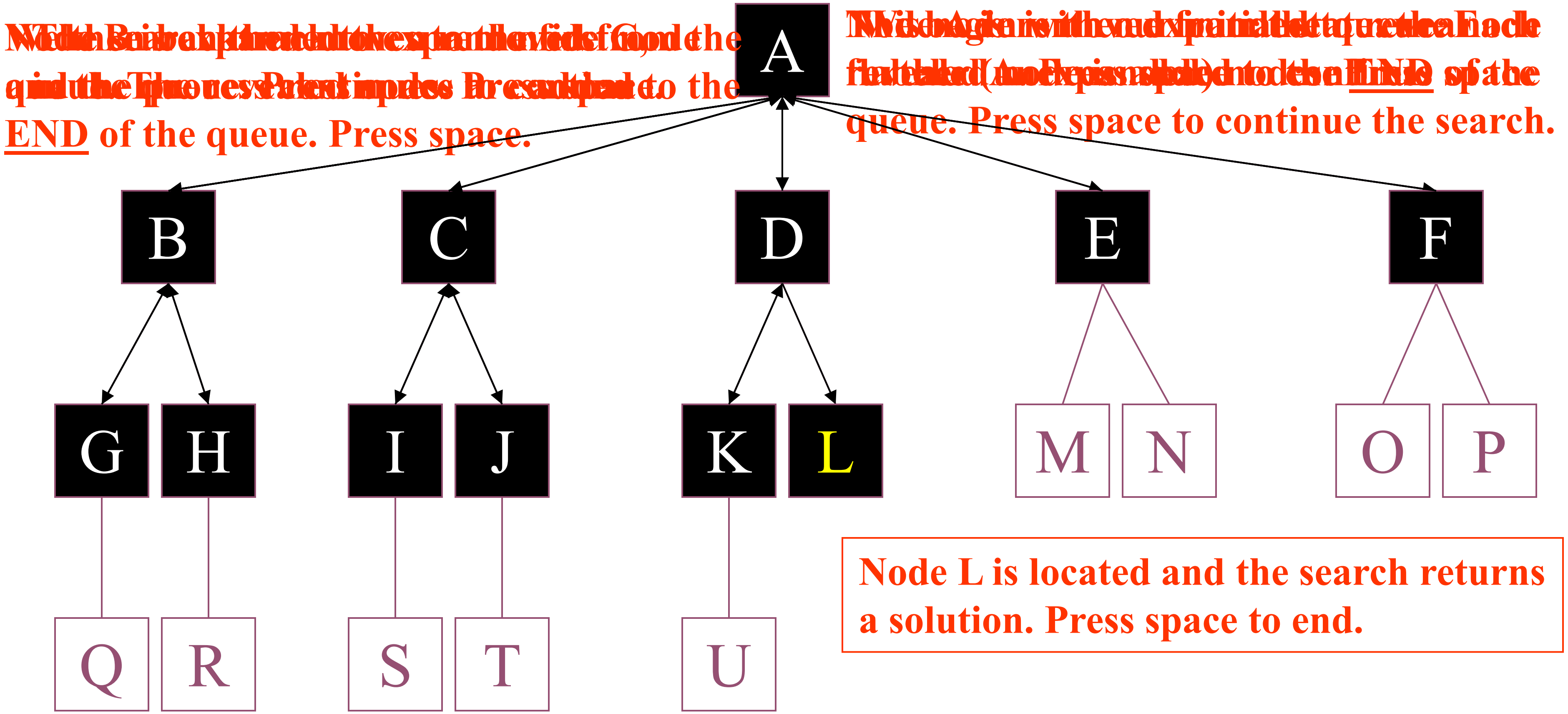
- Thus, new nodes (which are always deeper than their parents) go to the back of the queue, and old nodes, which are shallower than the new nodes, get expanded first.



The example node set



Press space to see a BFS of the example node set



[illegible] the [illegible] to the END of the queue. Press space.

[illegible] END of the queue. Press space to continue the search.

Node L is located and the search returns a solution. Press space to end.

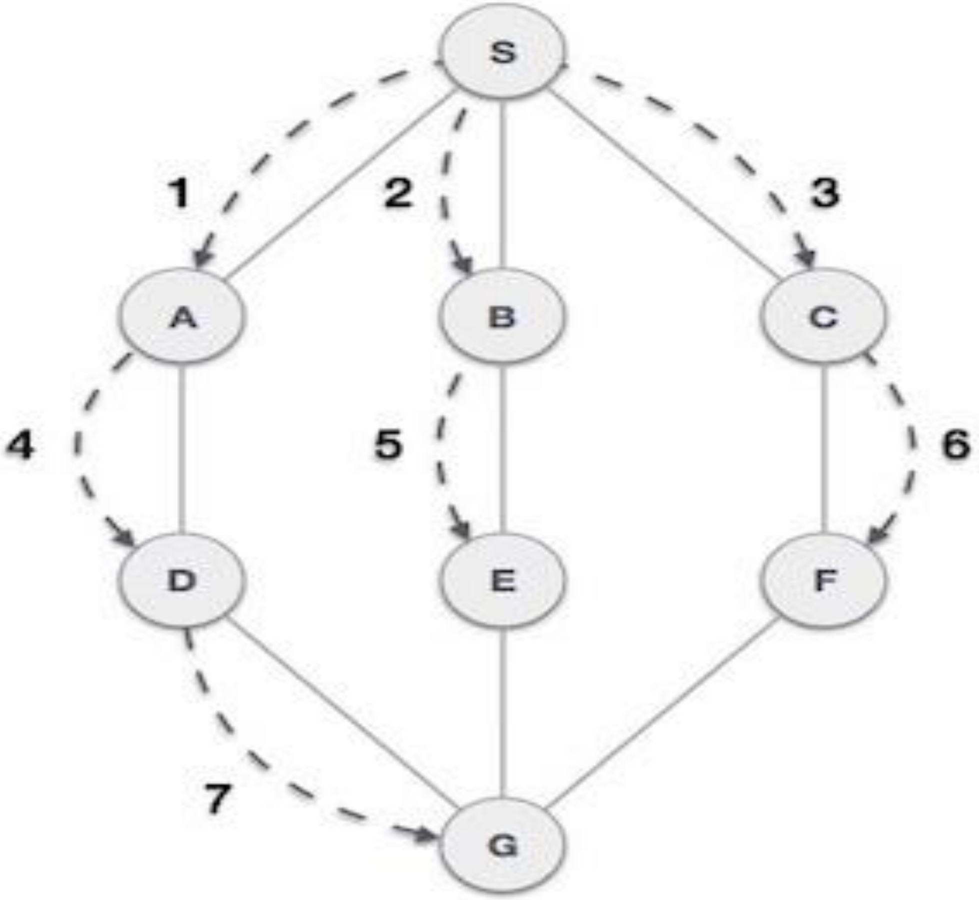
Press space to continue the search

Size of Queue: 0	Queue: Empty

Nodes expanded: 11	FINISHED SEARCH	Current level: 2

BREADTH-FIRST SEARCH PATTERN

Breadth First Search (Uninformed/Blind Search)



- 1. open = [S]; closed = []
- 2. open = [A,B,C]; closed = [S]
- 3. open = [B,C,D]; closed = [S,A]
- 4. open = [C,D,E]; closed = [S,A,B]
- 5. open = [D,E,F]; closed = [S,A,B,C]
- 6. open = [E,F,G]; closed = [S,A,B,C,D]
- 7. open = [F,G]; closed = [S,A,B,C,D,E]
- 8. open = [G] closed = [S,A,B,C,D,E,F]
- 9. Next is G goal is reached.

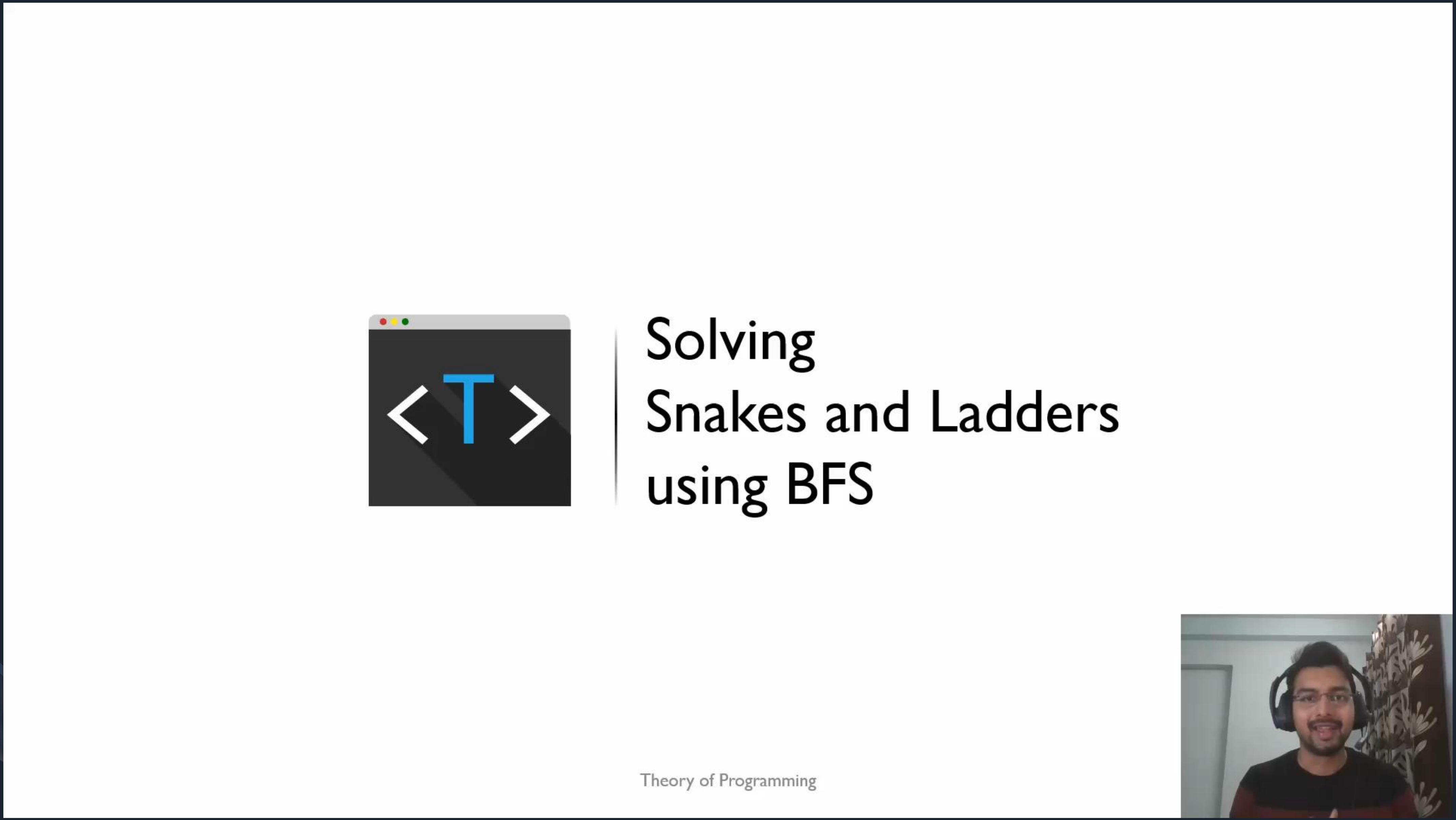
Breadth first search algorithm

```
• function BREADTH-FIRST-SEARCH(problem) returns a solution, or failure
•  node ←a node with STATE = problem.INITIAL-STATE, PATH-COST = 0
•  if problem.GOAL-TEST(node.STATE) then return SOLUTION(node)
•  frontier ←a FIFO queue with node as the only element
•  explored ←an empty set
•  loop do
•  if EMPTY?( frontier) then return failure
•  node←POP( frontier ) /* chooses the shallowest node in frontier */
•  add node.STATE to explored
•  for each action in problem.ACTIONS(node.STATE) do
•  child ←CHILD-NODE(problem, node, action)
•  if child .STATE is not in explored or frontier then
•  if problem.GOAL-TEST(child .STATE) then return SOLUTION(child )
•     frontier ←INSERT(child , frontier )
```

Applications of Breadth-First Search Algorithm

- Crawlers in Search Engines
- GPS Navigation systems
- Find the Shortest Path & Minimum Spanning Tree for an unweighted graph
- Broadcasting
- Peer to Peer Networking

Solving Snakes and Ladders using BFS



Solving
Snakes and Ladders
using BFS

Theory of Programming

https://www.youtube.com/watch?v=a3_QGPthIDU

Take care and Keep safe