



**CHANDIGARH  
UNIVERSITY**

Discover. Learn. Empower.

# **UNIVERSITY INSTITUTE OF ENGINEERING**

## **DEPARTMENT OF COMPUTER SCIENCE AND ENGG.**

Bachelor of Engineering (Computer Science & Engineering)

Principles of Artificial Intelligence (20CST-258)



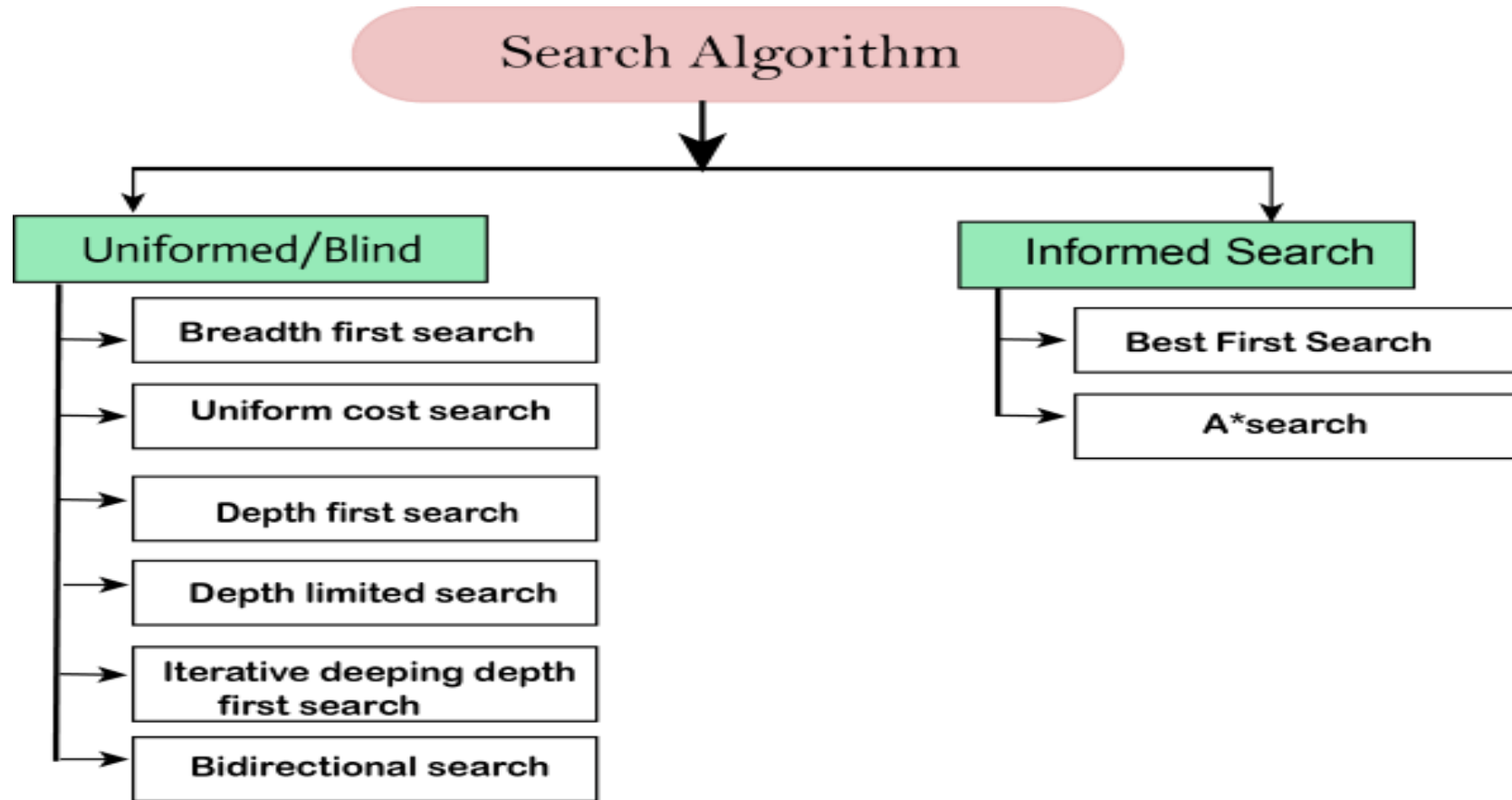
Breadth-first search

DISCOVER . **LEARN** . EMPOWER

# Outline

- Type of Search Strategies
  - Uninformed
    - Depth-first search
    - **Breadth-first search**
    - Iterative deepening depth-first search
    - Uniform cost search
    - Bidirectional Search
  - Informed

# Types of Search Algorithms



# The Uninformed Search

- Uninformed search algorithms have **no additional information on the goal node** other than the one provided in the problem definition.
- The plans to reach the goal state from the start state differ only by the order and/or length of actions.
- Uninformed search is also called **Blind search**.

These types of algorithms will have:

- A problem graph, containing the start node S and the goal node G.
- A strategy, describing the manner in which the graph will be traversed to get to G.
- A fringe, which is a data structure used to store all the possible states (nodes) that you can go from the current states.
- A tree that results while traversing to the goal node. □ A solution plan, which the sequence of nodes from S to G.

# Uninformed Search Types

- It can be divided into five main types:
  - Depth-first search
  - Breadth-first search
  - Iterative deepening depth-first search
  - Uniform cost search
  - Bidirectional Search

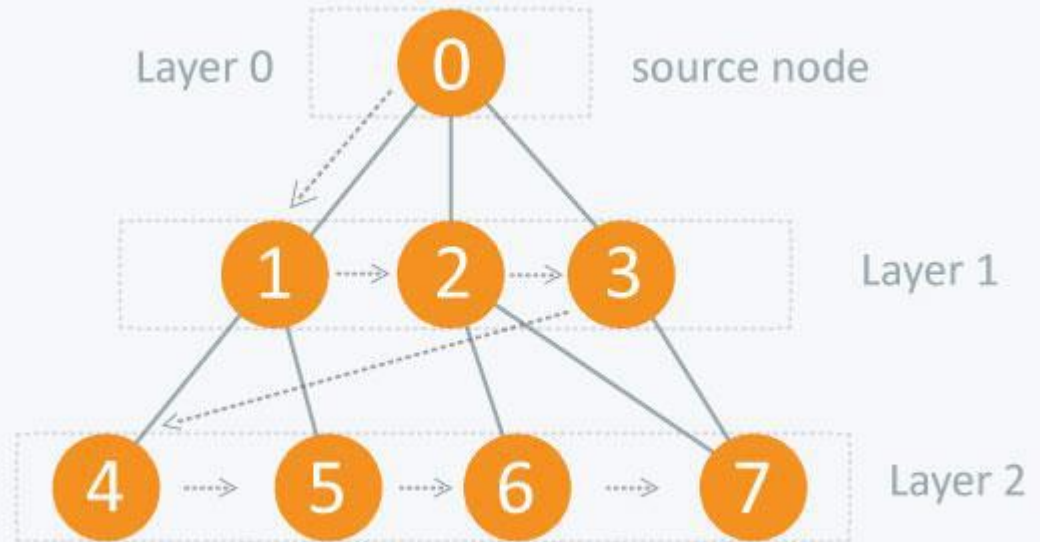
# Breadth First Search

- In Breadth First Search(BFS), the root node of the graph is expanded first, then all the successor nodes are expanded and then their successor and so on i.e. the nodes are expanded level wise starting at root level.
- 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.
- BFS algorithm starts searching from the root node of the tree and expands all successor nodes at the current level before moving to nodes of next level.
- The breadth-first search algorithm is an example of a general-graph search algorithm.
- Breadth-first search implemented using **FIFO queue data structure**.



# Example 1

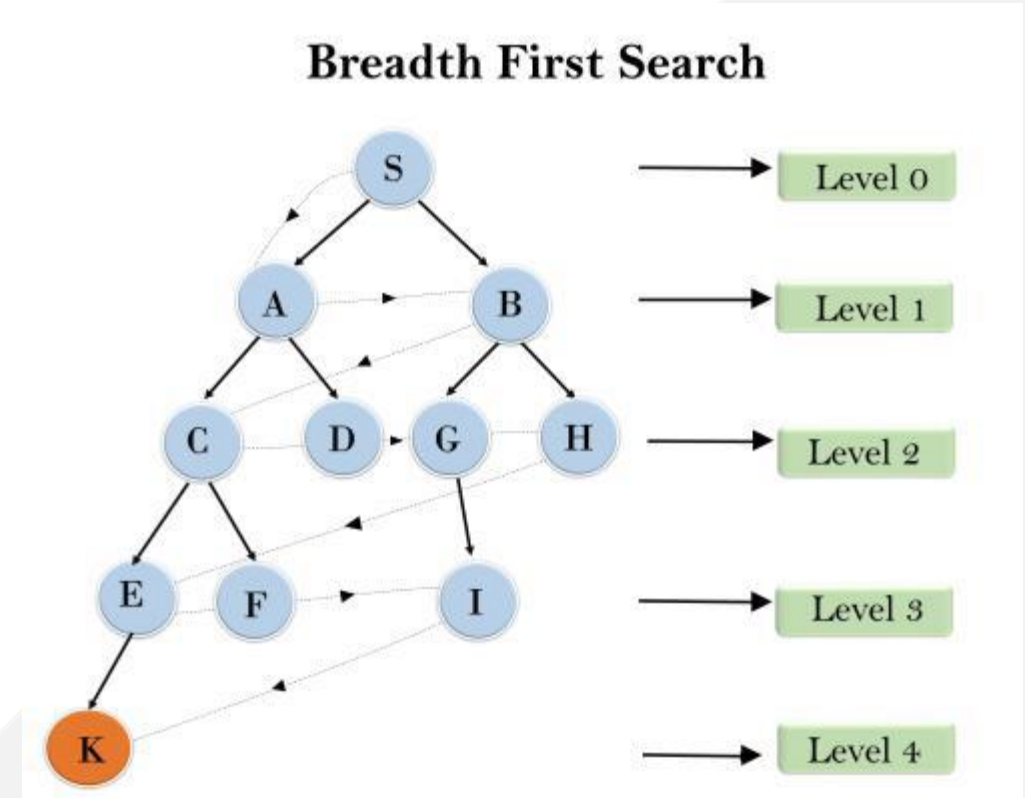
- Path will traverse will be 0-1-2-3-4-5-6-7
- Source node =0
- Destination node =7



# Example 2

- S= source node
- K= destination node
- Path will be according to BFS concept.

S----> A---->B---->C---->D---->G---->H---->  
E---->F---->I---->K





# Key Points

- **Time Complexity:** Time Complexity of BFS algorithm can be obtained by the number of nodes traversed in BFS until the shallowest Node.

$$T(b) = 1 + b^2 + b^3 + \dots + b^d = O(b^d)$$

Where,  $d$  = depth of shallowest solution

$b$  = node at every state.

- **Space Complexity:** Space complexity of BFS algorithm is given by the Memory size of frontier which is  $O(b^d)$ .
- **Completeness:** BFS is complete, which means if the shallowest goal node is at some finite depth, then BFS will find a solution.
- **Optimality:** BFS is optimal if path cost of all edges of tree/graph is same.



# Advantages of BFS

- In this procedure at any way it will find the goal.
- It does not follow a single unfruitful path for a long time. It finds the minimal solution in case of multiple paths.
- There is nothing like useless path in BFS, since it searches level by level.

# Disadvantages of BFS

- BFS consumes large memory space. Its time complexity is more.
- It has long pathways, when all paths to a destination are on approximately the same search depth.



**THANK YOU**