Search Strategy-1

Dr. Manjubala Bisi
(PhD, IIT Kharagpur)
Assistant Professor,
Dept. of CSE,
NIT Warangal

Introduction

- Search algorithms are one of the most important areas of Artificial Intelligence.
- In Artificial Intelligence, Search techniques are universal problem-solving methods.
- AI mostly used these search strategies or algorithms to solve a specific problem and provide the best result.

Search Algorithm Terminologies

- **Search:** Searching is a step by step procedure to solve a search-problem in a given search space. A search problem can have three main factors:
  - Search Space
  - Start State
  - Goal test

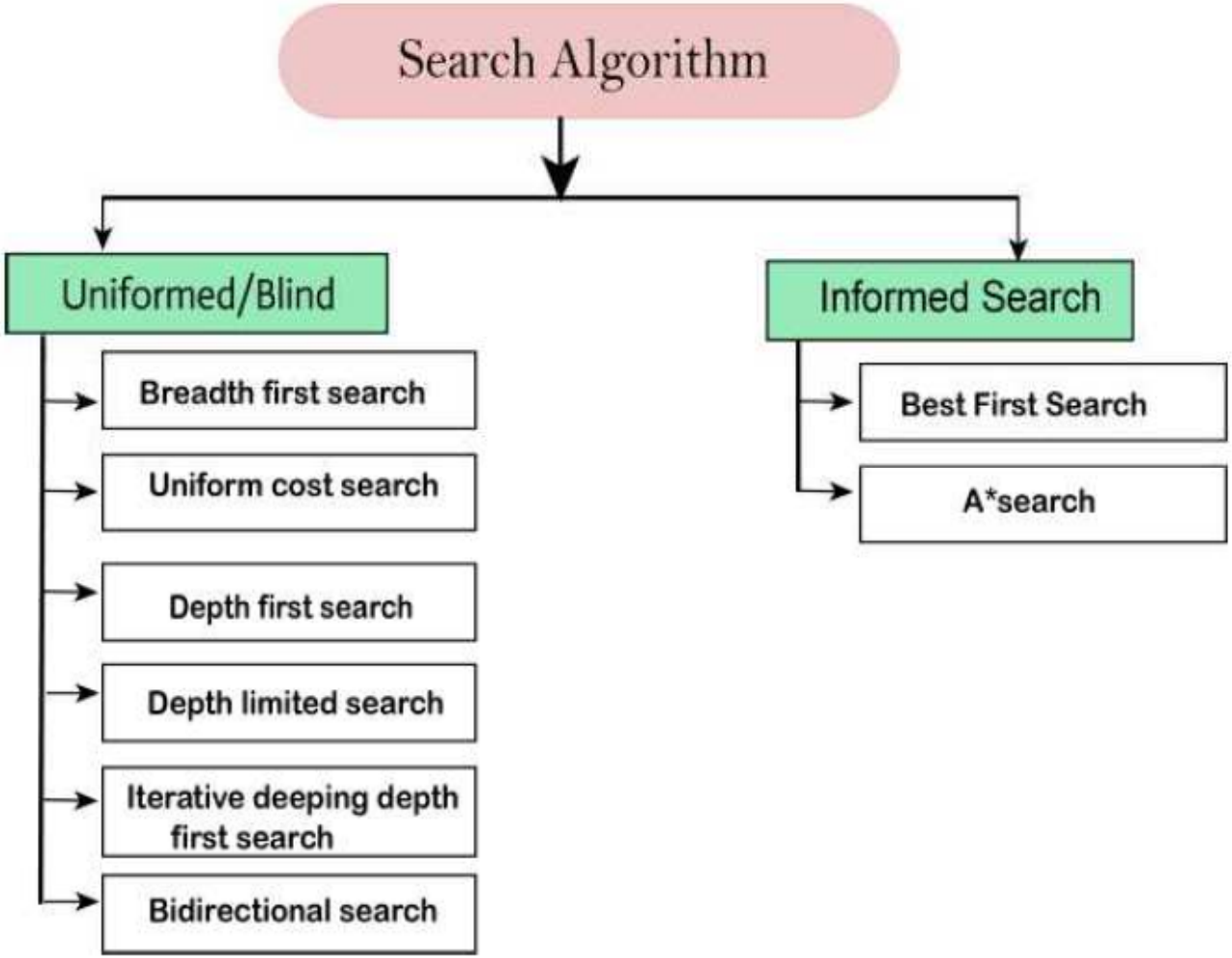
Search Algorithm Terminologies

- **Search tree**
- **Actions**
- **Transition model**
- **Path cost**
- **Solution**
- **Optimal solution**

Properties of Search Algorithms

- **Completeness**
- **Optimality**
- **Time Complexity**
- **Space Complexity**

Types of Search Algorithms



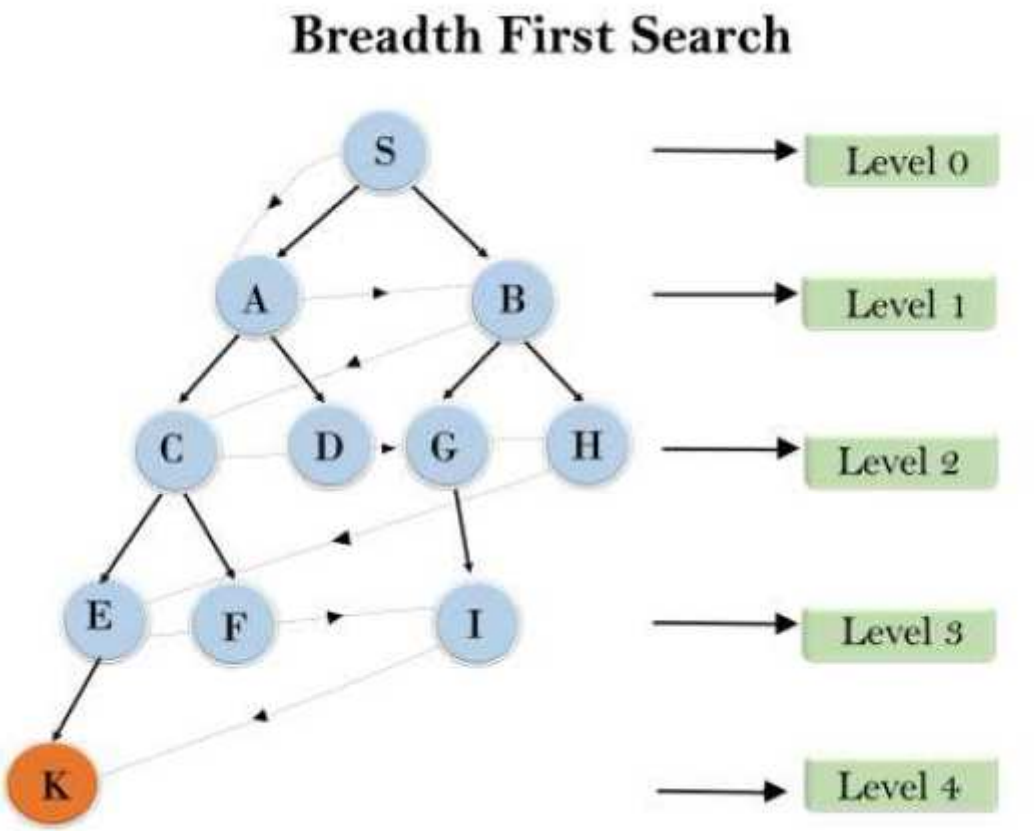
Uninformed Search Algorithms

- Uninformed search is a class of general-purpose search algorithms which operates in brute force-way.
- Uninformed search algorithms do not have additional information about state or search space other than how to traverse the tree.
- It is also called **blind search.**

Breadth-first Search

- Breadth-first search is the most common search strategy for traversing a tree or graph. This algorithm searches breadth wise in a tree or graph.
- It starts searching from the root node of the tree and expands all successor node at the current level before moving to nodes of next level.
- It is implemented using Queue data structure.

Example of Breadth-first Search



Breadth-first Search

Advantages:

- BFS will provide a solution if any solution exists.
- If there are more than one solutions for a given problem, then BFS will provide the minimal solution which requires the least number of steps.

Disadvantages:

- It requires lots of memory since each level of the tree must be saved into memory to expand the next level.
- It needs lots of time if the solution is far away from the root node.

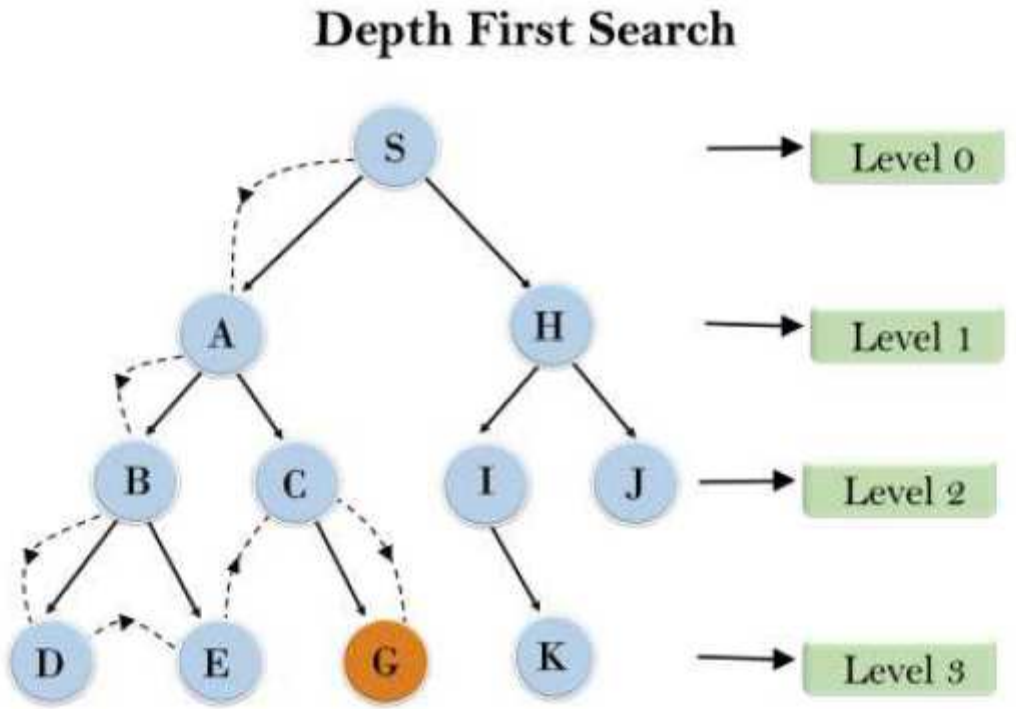
Breadth-first Search

- **Time Complexity:** T (b) = $O(b^d)$
- **Space Complexity:** Space complexity is $O(b^d)$.
- **Completeness:** BFS is complete.
- **Optimality:** BFS is optimal if path cost is a non-decreasing function of the depth of the node.

Depth-first Search

- Depth-first search is a recursive algorithm for traversing a tree or graph data structure.
- It is called the depth-first search because it starts from the root node and follows each path to its greatest depth node before moving to the next path.
- It uses a stack data structure for its implementation.

Example of Depth-first Search



Depth-first Search

Advantages:

- DFS requires very less memory as it only needs to store a stack of the nodes on the path from root node to the current node.
- It takes less time to reach to the goal node than BFS algorithm (if it traverses in the right path).

Disadvantages:

- There is the possibility that many states keep re-occurring, and there is no guarantee of finding the solution.
- DFS algorithm goes for deep down searching and sometime it may go to the infinite loop.

Depth-first Search

- **Time Complexity:**

  $T(b)=O(b^d)$

- **Space Complexity:** DFS algorithm needs to store only single path from the root node, which is O(bd).
- **Completeness**: DFS search algorithm is complete.
- **Optimal**: DFS search algorithm is non-optimal

.

Depth-Limited Search Algorithm

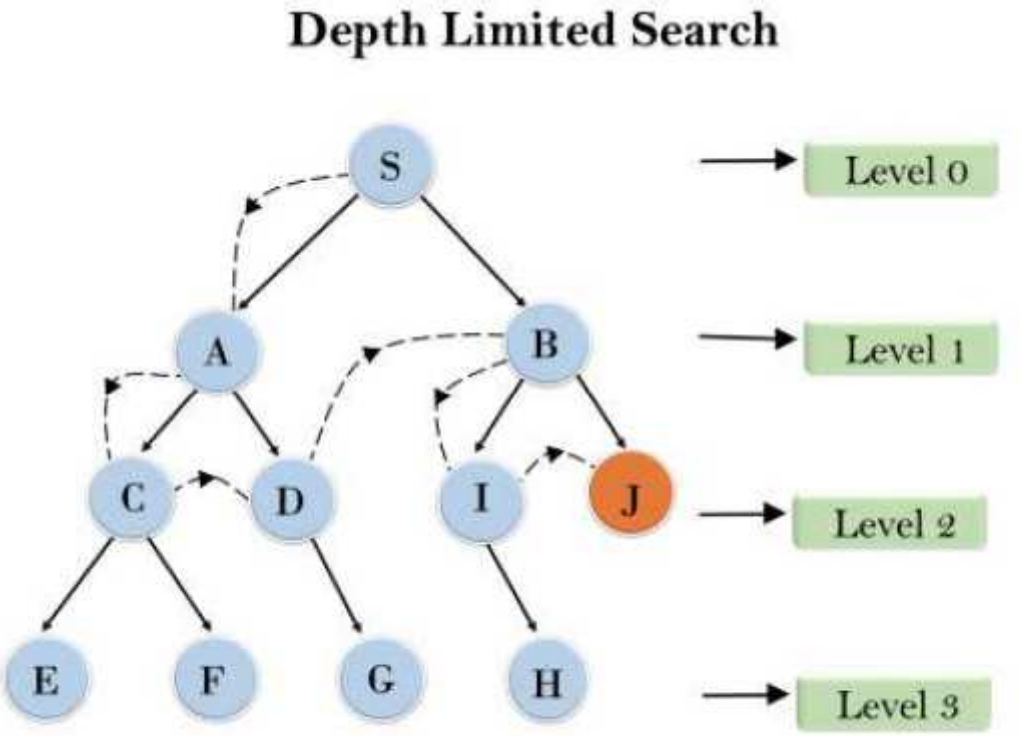
- A depth-limited search algorithm is similar to depth-first search with a predetermined limit.
- It can solve the drawback of the infinite path in the Depth-first search.
- In this algorithm, the node at the depth limit will treat as it has no successor nodes further.

Depth-Limited Search Algorithm

Depth-limited search can be terminated with two conditions of failure:

- **Standard failure value:** It indicates that problem does not have any solution.
- **Cutoff failure value:** It defines no solution for the problem within a given depth limit.

Example of Depth-Limited Search



Depth-Limited Search

Advantages:

- Depth-limited search is Memory efficient.

Disadvantages:

- Depth-limited search also has a disadvantage of incompleteness.
- It may not be optimal if the problem has more than one solution.

Depth-Limited Search

- **Time Complexity:**

  Time complexity of DLS algorithm is $O(b^{\ell})$.

- **Space Complexity:**

  Space complexity of DLS algorithm is $O(b \times \ell)$.

- **Completeness:**

  It is complete if the solution is above the depth-limit.

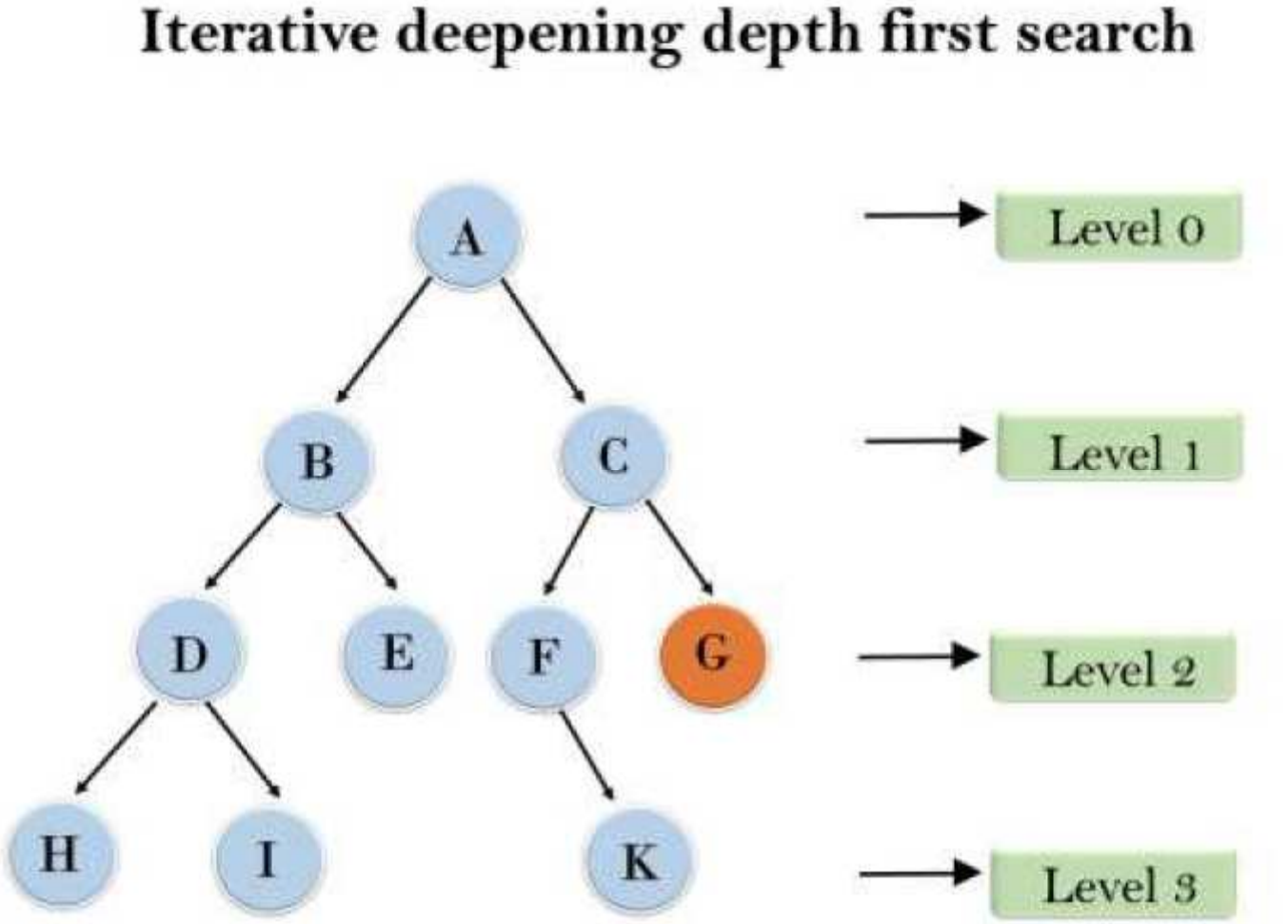
- **Optimal:** It is a special case of DFS, and it may not optimal.

Iterative deepening depth-first Search

- This search algorithm finds out the best depth limit and does it by gradually increasing the limit until a goal is found.

- It performs depth-first search up to a certain "depth limit", and it keeps increasing the depth limit after each iteration until the goal node is found.

Example of Iterative deepening depth-first Search



Iterative deepening depth-first Search

Advantages:

- It combines the benefits of BFS and DFS search algorithm in terms of fast search and memory efficiency.

Disadvantages:

- The main drawback of this search is that it repeats all the work of the previous phase.

Iterative deepening depth-first Search

1'st Iteration-----> A
2'nd Iteration----> A, B, C
3'rd Iteration------>A, B, D, E, C, F, G
4'th Iteration------>A, B, D, H, I, E, C, F, K, G

Iterative deepening depth-first Search

- **Time Complexity:**

  $T(b)=O(b^d)$.

- **Space Complexity:**

  The space complexity is O(bd).

- **Completeness:**

  It is complete is if the branching factor is finite.

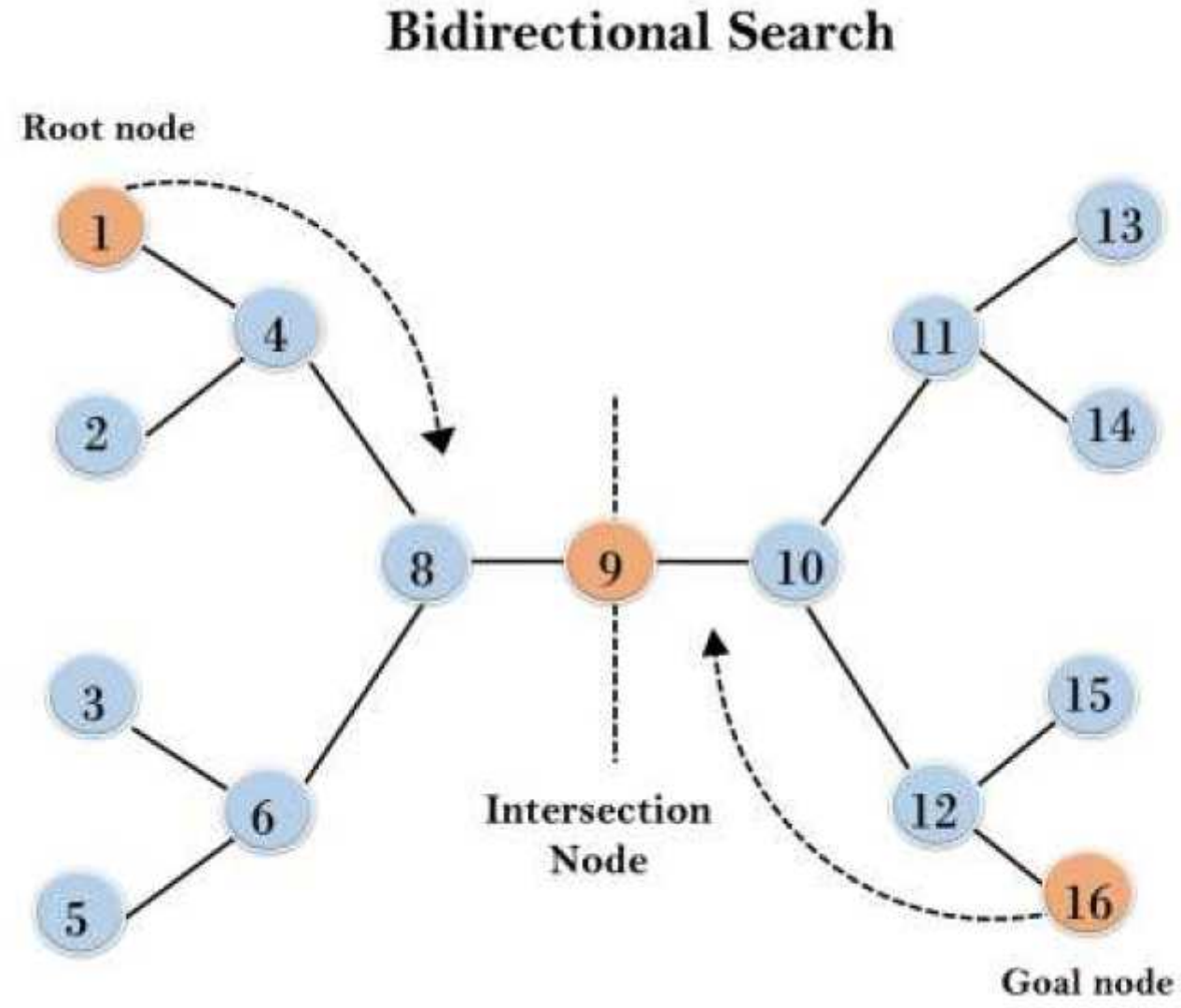
- **Optimal:**

  It is optimal if path cost is a non- decreasing function of the depth of the node.

Bidirectional Search Algorithm

- Bidirectional search algorithm runs two simultaneous searches, one form initial state called as forward-search and other from goal node called as backward-search, to find the goal node.
- It replaces one single search graph with two small subgraphs in which one starts the search from an initial vertex and other starts from goal vertex.
- The search stops when these two graphs intersect each other.
- Bidirectional search can use search techniques such as BFS, DFS, DLS, etc.

Example of Bidirectional Search Algorithm



Bidirectional Search Algorithm

Advantages:

- Bidirectional search is fast.
- Bidirectional search requires less memory

Disadvantages:

- Implementation of the bidirectional search tree is difficult.
- In bidirectional search, one should know the goal state in advance.

Bidirectional Search Algorithm

- **Time Complexity:**

  Time complexity using BFS is $O(b^d)$.

- **Space Complexity:**

  Space complexity of is $O(b^d)$.

- **Completeness:**

  It is complete if we use BFS in both searches.

- **Optimal:**

  It is optimal.

Informed search algorithms

- Greedy search
- Alpha-beta pruning search
- A* search

Local and Optimization algorithms for Search

- Hill climbing
- Simulated annealing
- Genetic algorithm
- Particle swarm optimization
- Ant-colony optimization
- Teaching learning based optimization

Thank You

?