UNIT II

Introduction

Contents

Unit II	Problem-solving	07 Hours
Solving Problems by Searching, Problem-Solving Agents, Example Problems, Search Algorithms, Uninformed Search Strategies, Informed (Heuristic) Search Strategies, Heuristic Functions, Search in Complex Environments, Local Search and Optimization Problems.		
#Exemplar/Case Studies	4th Industrial Revolution Using AI, Big Data And Robotics	
***Mapping of Course Outcomes for Unit II**	CO2, CO4	

Depth-Limited Search Algorithm

- ✔ A depth-limited search algorithm is similar to depth-first search with a predetermined limit.
- ✔ Depth-limited search 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 can be terminated with two Conditions of failure:

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

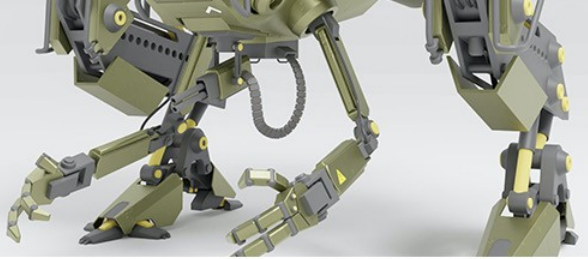
Depth-Limited Search Algorithm

✔ **Advantages:**
Depth-limited search is Memory efficient.

✔ **Disadvantages:**

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

Depth-Limited Search Algorithm

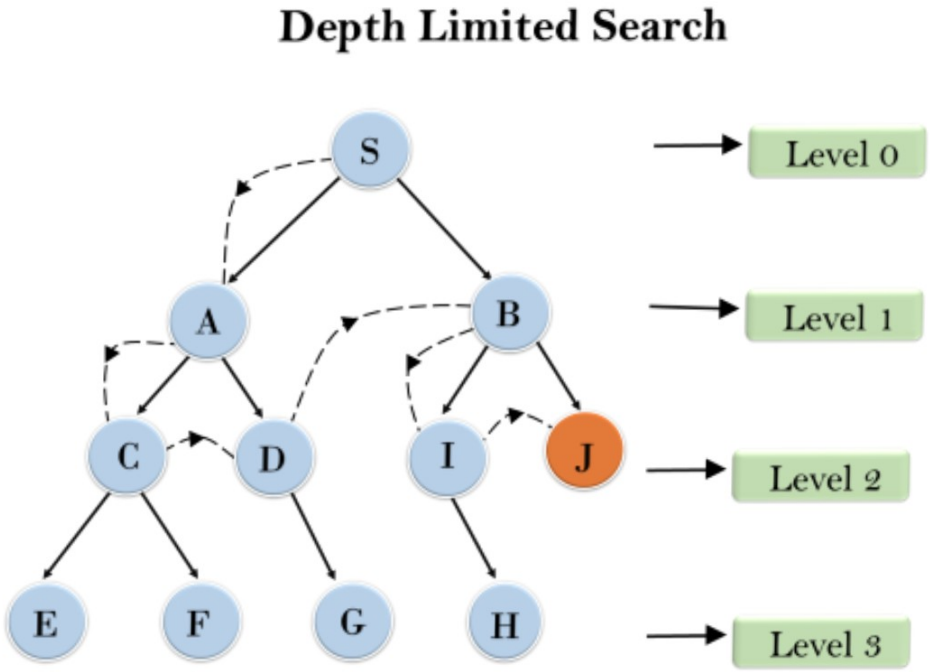


Completeness: DLS search algorithm is complete if the solution is above the depth-limit.

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

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

Optimal: Depth-limited search can be viewed as a special case of DFS, and it is also not optimal even if $\ell > d$.



Iterative deepeningdepth-first Search

- The iterative deepening algorithm is a combination of DFS and BFS algorithms. This search algorithm finds out the best depth limit and does it by gradually increasing the limit until a goal is found.
- This algorithm 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.
- This Search algorithm combines the benefits of Breadth-first search's fast search and depth-first search's memory efficiency.
- The iterative search algorithm is useful uninformed search when search space is large, and depth of goal node is unknown.

Iterative deepeningdepth-first Search

- **Advantages:**

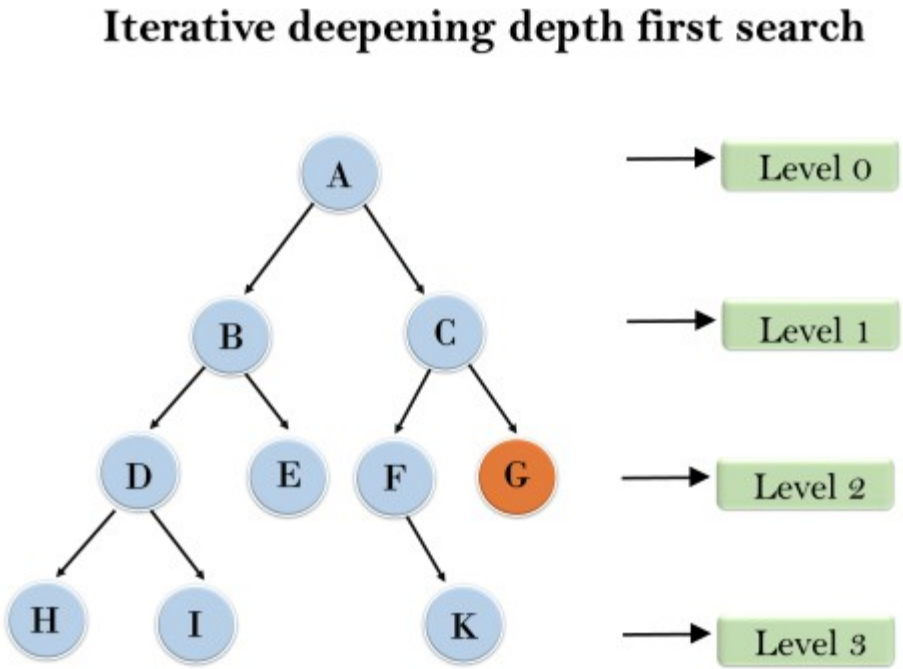
  Itcombines the benefits of BFS and DFS search algorithm in terms of fast search and memory efficiency.

- **Disadvantages:**

  The main drawback of IDDFS is that it repeats all the work of the previous phase.

Iterative deepeningdepth-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
In the fourth iteration, the algorithm will find the goal node.



Iterative deepeningdepth-first Search

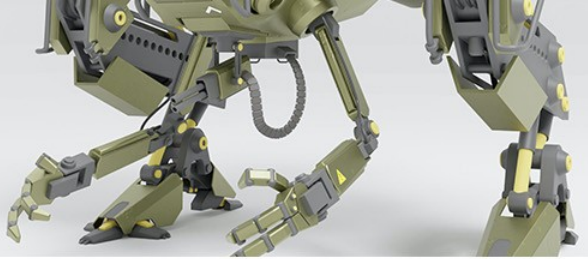
- **Completeness:**
  This algorithm is complete is ifthe branching factor is finite.

- **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 O(bd).

- **Optimal:**
  IDDFS algorithm 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.
- Bidirectional search 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.**

Bidirectional Search Algorithm:

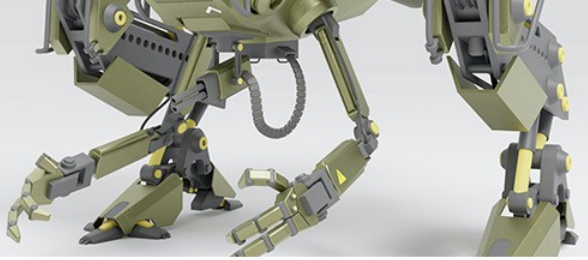
- **Advantages:**

1. Bidirectional search is fast.
2. Bidirectional search requires less memory

- **Disadvantages:**

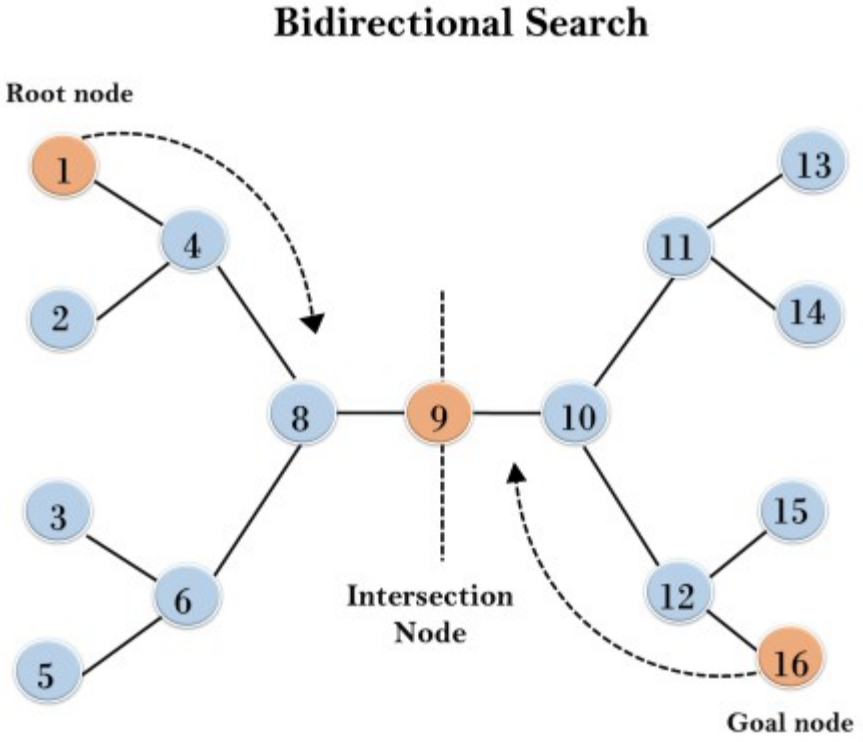
1. Implementation of the bidirectional search tree is difficult.
2. In bidirectional search, one should know the goal state in advance.

Bidirectional Search Algorithm:



Example:
In the below search tree, bidirectional search algorithm is applied. This algorithm divides one graph/tree into two sub-graphs. It starts traversing from node 1 in the forward direction and starts from goal node 16 in the backward direction.

The algorithm terminates at node 9 where two searches meet.



Bidirectional Search Algorithm:

- **Completeness:** Bidirectional Search is complete if we use BFS in both searches.
- **Time Complexity:** Time complexity of bidirectional search using BFS is $O(b^d)$.
- **Space Complexity:** Space complexity of bidirectional search is $O(b^d)$.
- **Optimal:** Bidirectional search is Optimal.