Search Algorithm in Artificial Intelligence

Search

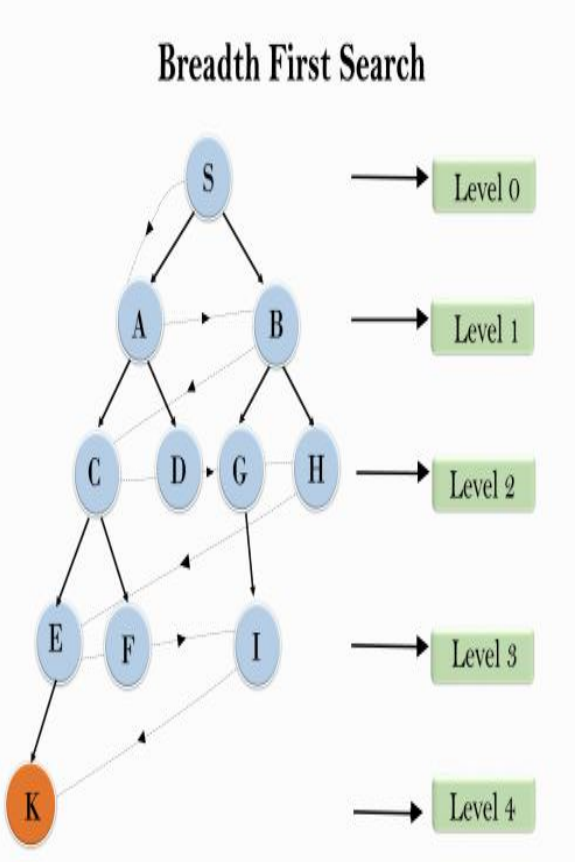
- Searching is a step by step method to solve a search-problem in a specified search space. A search problem can have three main factors:
  - Search Space
  - Start State
  - Goal test
  - **Properties of Search Algorithms-**
    - Completeness
    - Optimality
    - Time Complexity
    - Space Complexity

Types of search algorithms

1. **Uninformed/Blind Search:- like**
   - Breadth-first search
   - Depth-first search
   - Depth-Limited search
   - Uniform cost search
   - Iterative deepening depth-first search
   - Bidirectional Search
2. **Informed Search:- like**
   - Greedy Search
   - A* Search

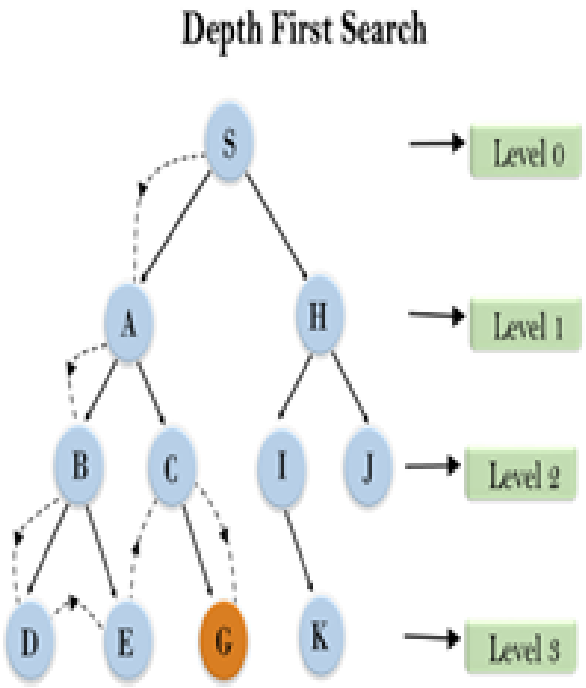
Breadth-first Search

- BFS is the most common search for traversing a tree or graph. This algorithm searches breadthwise in a tree or graph, so it is called breadth-first search.
- In BFS search starts from root node and then before moving to next level all successor node from current level is expand.
- It is an example of a general-graph search algorithm.
- it uses queue data structure.



Depth-first Search

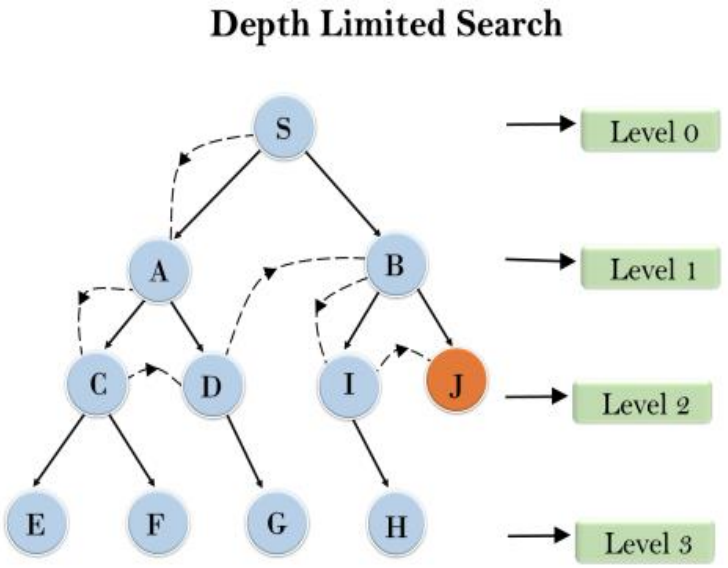
- Depth-first search is a recursive algorithm for traversing a tree or graph data structure.
- It is called the depth-first search because it starts from the root node and follows each path to its greatest depth node before moving to the next path.
- DFS uses a stack data structure for its implementation.
- Backtracking is an algorithm technique for finding all possible solutions using recursion.



Depth-Limited Search Algorithm

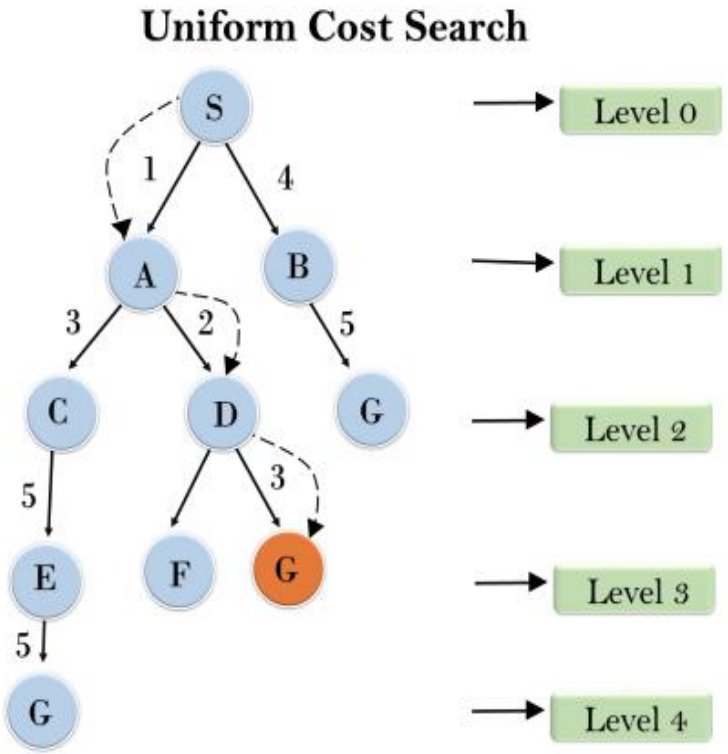
[DFS + Predetermined limit]

- It can solve the problem of infinite path in the Depth-first search.
- The node at the depth limit will treat as it has no successor nodes further.



Uniform-cost Search Algorithm

- In this algorithm at every state the path with the least cost is selected.



Informed Search Algorithms

- The informed search algorithm is more useful for large search space. Informed search algorithm uses the idea of heuristic, so it is also called Heuristic search.
- **Heuristics function:** Heuristic is a function which is used in Informed Search, and it finds the most promising path. It takes the current state of the agent as its input and produces the estimation of how close agent is from the goal. The heuristic method, however, might not always give the best solution, but it guaranteed to find a good solution in reasonable time. Heuristic function estimates how close a state is to the goal. It is represented by h(n), and it calculates the cost of an optimal path between the pair of states. The value of the heuristic function is always positive.
- **Admissibility of the heuristic function is given as:**

  $$h(n) <= h^*(n)$$

- **Here h(n) is heuristic cost, and h*(n) is the estimated cost. Hence heuristic cost should be less than or equal to the estimated cost.**

1. Best-first Search Algorithm (Greedy Search)

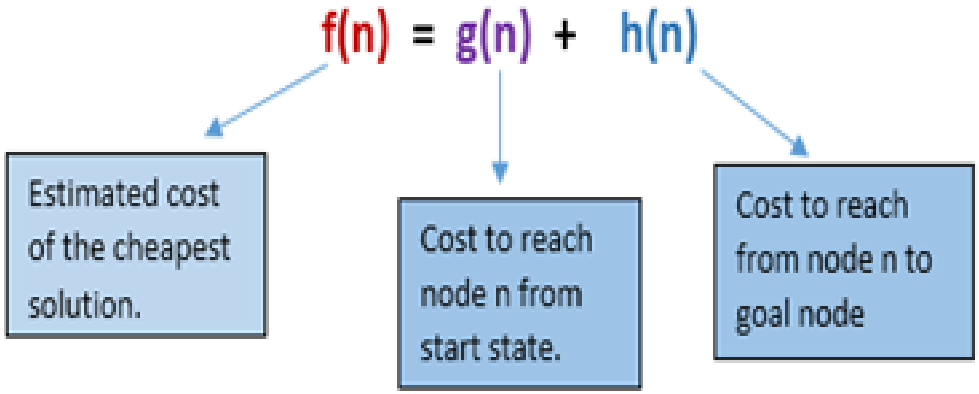
- Greedy best-first search algorithm always selects the path which appears best at that moment. It is the combination of depth-first search and breadth-first search algorithms. It uses the heuristic function and search. Best-first search allows us to take the advantages of both algorithms. With the help of best-first search, at each step, we can choose the most promising node. In the best first search algorithm, we expand the node which is closest to the goal node and the closest cost is estimated by heuristic function, i.e.

  $$f(n) = g(n).$$

- Were, h(n)= estimated cost from node n to the goal.
- The greedy best first algorithm is implemented by the priority queue.

2. A* Search Algorithm

- In A* search algorithm, we use search heuristic as well as the cost to reach the node. Hence we can combine both costs as following, and this sum is called as a **fitness number**.



- At each point in the search space, only those node is expanded which have the lowest value of f(n), and the algorithm terminates when the goal node is found.
- A* search algorithm is optimal and complete.

Hill Climbing Algorithm

- It is a local search algorithm which continuously moves in the direction of increasing elevation/value to find the peak of the mountain or best solution to the problem. It terminates when it reaches a peak value where no neighbor has a higher value.
- **Features of Hill Climbing Algorithm:**
  - **Generate and Test variant**
  - **Greedy approach**
  - **No backtracking**
- **Simple Hill Climbing**: It only evaluates the neighbor node state at a time and selects the first one which optimizes current cost and set it as a current state.
  - Less time consuming
  - Less optimal solution and the solution is not guaranteed

Problems in Hill Climbing Algorithm

- **Local Maximum**- A local maximum is a peak state in the landscape which is better than each of its neighboring states, but there is another state also present which is higher than the local maximum.
- **Plateau**- A plateau is the flat area of the search space in which all the neighbor states of the current state contains the same value, because of this algorithm does not find any best direction to move. A hill-climbing search might be lost in the plateau area.
- **Ridges**- A ridge is a special form of the local maximum. It has an area which is higher than its surrounding areas, but itself has a slope, and cannot be reached in a single move.