UNIVERSITY INSTITUTE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGG.

Bachelor of Engineering (Computer Science & Engineering)

Principles of Artificial Intelligence (20CST-258)

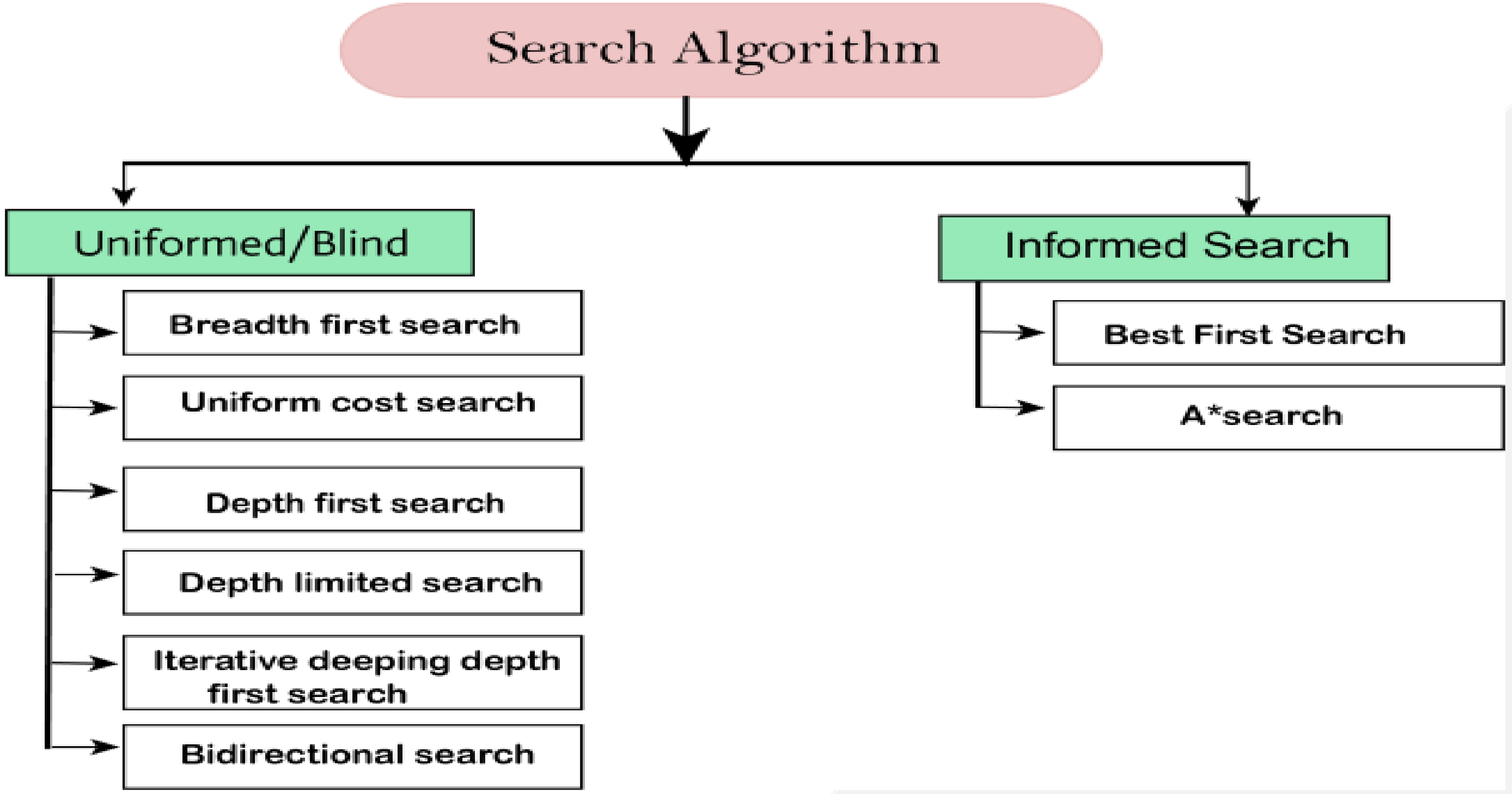
DISCOVER . LEARN . EMPOWER

Informed Search

Outline

- Type of Search Strategies
- Pure Heuristic Search
    - Best First Search Algorithm(Greedy search)
    - A* Search Algorithm
- Hill Climbing:
    - Simple hill Climbing
    - Steepest-Ascent hill-climbing
    - Stochastic hill Climbing

Types of Search Algorithms



The Informed Search

- Informed search methods use knowledge about the problem domain and choose promising operators first.
- These heuristic search methods use heuristic functions to evaluate the next state towards the goal state.

  For finding a solution, by using the heuristic technique, one should carry out the following steps:-

  1. Add domain: specific information to select what is the best path to continue searching along
  2. Define a heuristic function h(n): that estimates the ‗goodness‘ of a node n. Specifically, h(n) = estimated cost(or distance) of minimal cost path from n to a goal state.
  3. The term, heuristic means "serving to aid discovery" and is an estimate, based on domain specific information that is computable from the current state description of how close we are to a goal.

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.
- 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.
- 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.

Characteristics of heuristic search

- Heuristics are knowledge about domain, which help search and reasoning in its domain.
- Heuristic search incorporates domain knowledge to improve efficiency over blind search.
- Heuristic is a function that, when applied to a state, returns value as estimated merit of state, with respect to goal.
- Heuristic evaluation function estimates likelihood of given state leading to goal state.
- Heuristic search function estimates cost from current state to goal, presuming function is efficiency

Pure Heuristic Search

- Pure heuristic search is the simplest form of heuristic search algorithms.
- It expands nodes based on their heuristic value h(n). It maintains two lists, OPEN and CLOSED list.
- In the CLOSED list, it places those nodes which have already expanded and in the OPEN list, it places nodes which have yet not been expanded.
- On each iteration, each node n with the lowest heuristic value is expanded and generates all its successors and n is placed to the closed list. The algorithm continues unit a goal state is found.

Heuristic Search

- **Pure Heuristic Search**
  - In the informed search, there are two main algorithms:
    - Best First Search Algorithm(Greedy search)
    - A* Search Algorithm
- **Hill Climbing:**
    - Simple hill Climbing
    - Steepest-Ascent hill-climbing
    - Stochastic hill Climbing

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.

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

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

Best First Search Algorithm(Greedy search)

- Best first search algorithm:
  - **Step 1:** Place the starting node into the OPEN list.
  - **Step 2:** If the OPEN list is empty, Stop and return failure.
  - **Step 3:** Remove the node n, from the OPEN list which has the lowest value of h(n), and places it in the CLOSED list.
  - **Step 4:** Expand the node n, and generate the successors of node n.
  - **Step 5:** Check each successor of node n, and find whether any node is a goal node or not. If any successor node is goal node, then return success and terminate the search, else proceed to Step 6.
  - **Step 6:** For each successor node, algorithm checks for evaluation function f(n), and then check if the node has been in either OPEN or CLOSED list. If the node has not been in both list, then add it to the OPEN list.
  - **Step 7:** Return to Step 2.

Advantages

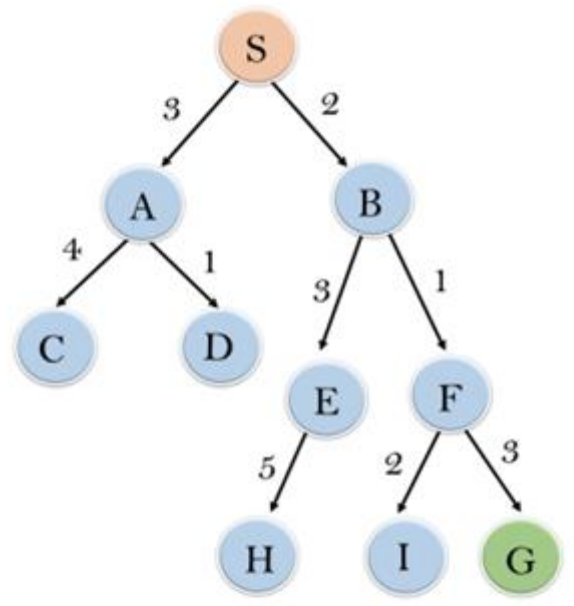
- Best first search can switch between BFS and DFS by gaining the advantages of both the algorithms.
- This algorithm is more efficient than BFS and DFS algorithms.

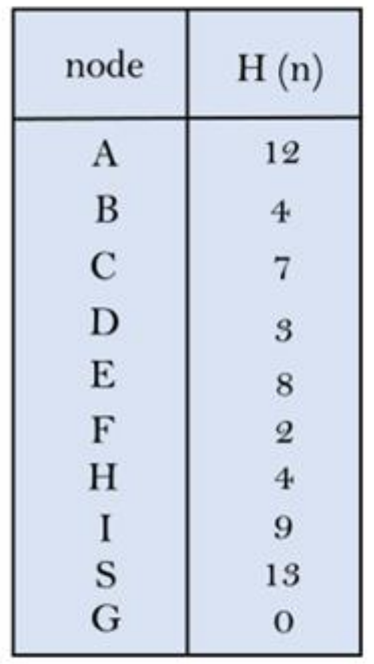
Disadvantages:

- Best first search can switch between BFS and DFS by gaining the advantages of both the algorithms.
- This algorithm is more efficient than BFS and DFS algorithms.
- It can behave as an unguided depth-first search in the worst case scenario.
- It can get stuck in a loop as DFS.
- This algorithm is not optimal.

Example

- Consider the below search problem, and we will traverse it using greedy best-first search.
- At each iteration, each node is expanded using evaluation function f(n)=h(n) , which is given in the below table.

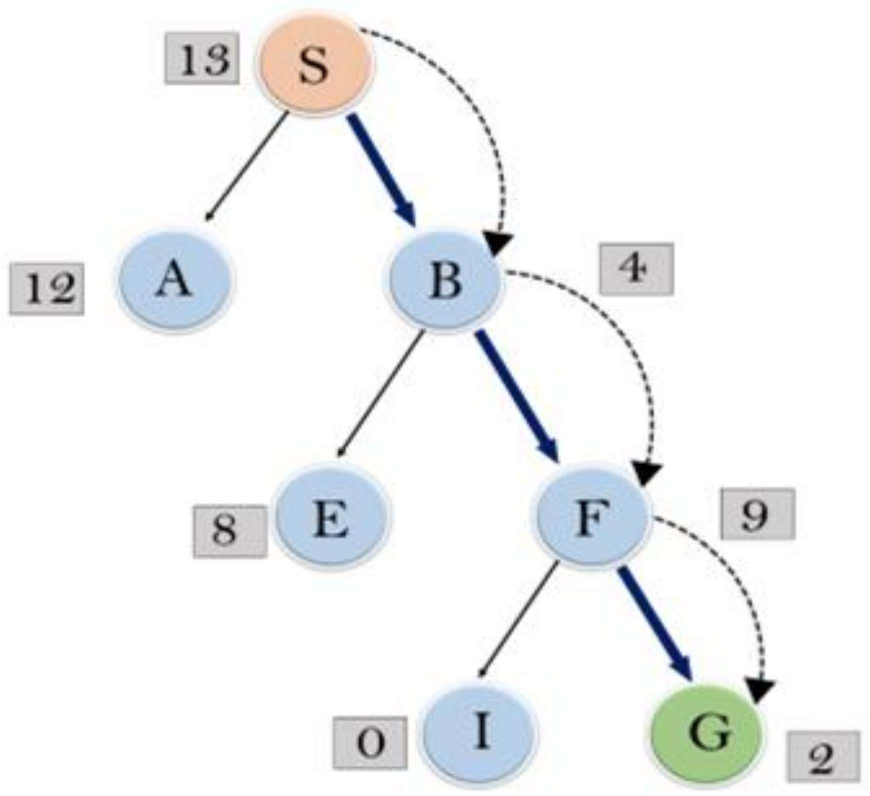




node	H (n)
A	12
B	4
C	7
D	3
E	8
F	2
H	4
I	9
S	13
G	0

Example

- In this search example, we are using two lists which are **OPEN** and **CLOSED** Lists. Following are the iteration for traversing the above example.



Expand the nodes of S and put in the CLOSED list

Example

- **Initialization:** Open [A, B], Closed [S]
- **Iteration 1:** Open [A], Closed [S, B]
- **Iteration2:** Open [E, F, A], Closed [S, B]
  : Open [E, A], Closed [S, B, F]
- **Iteration 3:** Open [I, G, E, A], Closed [S, B, F]
  : Open [I, E, A], Closed [S, B, F, G]
- Hence the final solution path will be: **S----> B----->F----> G**

Key Points

- **Time Complexity:** The worst case time complexity of Greedy best first search is O(b^m).
- **Space Complexity:** The worst case space complexity of Greedy best first search is O(b^m). Where, m is the maximum depth of the search space.
- **Complete:** Greedy best-first search is also incomplete, even if the given state space is finite.
- **Optimal:** Greedy best first search algorithm is not optimal.

A* Search Algorithm:

- A* search is the most commonly known form of best-first search.
- It uses heuristic function h(n), and cost to reach the node n from the start state g(n).
- It has combined features of UCS and greedy best-first search, by which it solve the problem efficiently.
- A* search algorithm finds the shortest path through the search space using the heuristic function. This search algorithm expands less search tree and provides optimal result faster.

THANK YOU