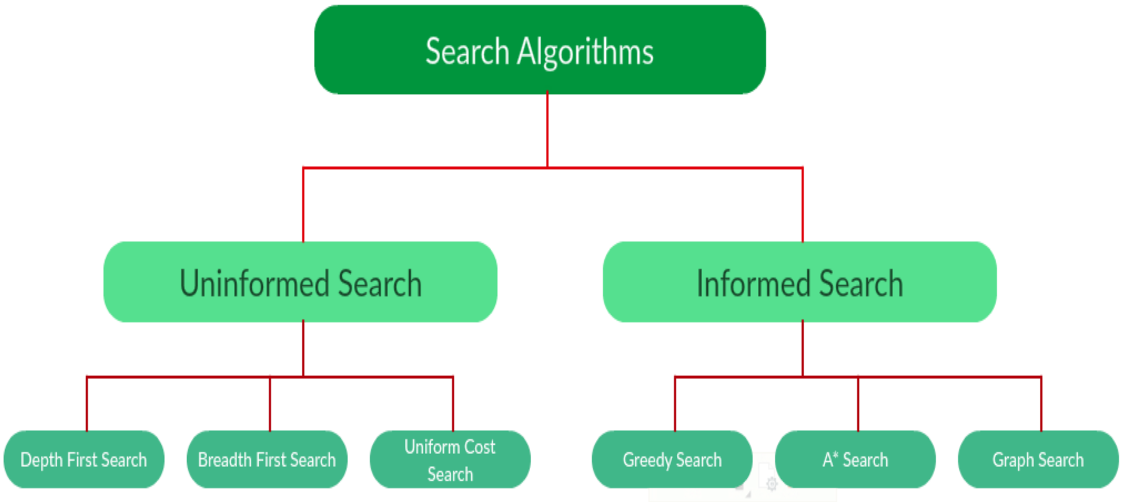
Applications of Uninformed and Informed Search Algorithms

Manal Anetallah Alsahafi

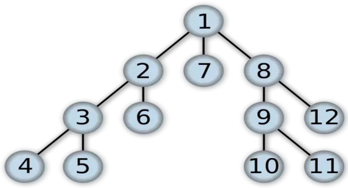
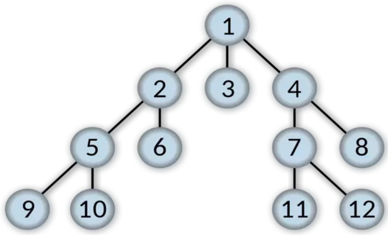
Introduction

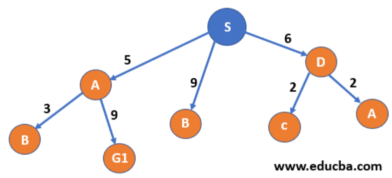
A search algorithm defines a sequence of actions (solutions) that reaches the goal [1]. Without these algorithms, AI agents cannot implement search functions to find the best solution for the problem from all possible alternatives or solutions. Of the many powerful search algorithms, these are the six fundamental search algorithms in AI, divided into two categories, see figure 1.



(Image taken from Search Algorithms in AI- geeksforgeeks)

Uninformed Search (also called blind or brute-force) search algorithm generates the graph without using any information about the number of steps to reach from the current state to the goal state. On the other hand, an **informed search** algorithm generates the graph using information from the *heuristic function* about how close a current state is to the goal state [2].

Search algorithm	Strategy	Applications
Depth-First-Search **(DFS)**	Expand the shallowest unexpanded node  source: Depth-first search - Wikipedia	• Solving puzzles with only one solution, such as mazes. • Path finding.
Breadth-First-Search **(BFS)**	Expand the deepest unexpanded node.  source: Breadth-first search - Wikipedia	• Unweighted graph minimum spanning tree. • Find the shortest path if the weight of each edge between any two nodes is the same. • Computer Networks in Peer to peer (P2P) applications and Broadcasting. • Social Networking Websites to find any random person in the world by traversing nodes. • GPS Navigation systems to reach from one place to another. • Web crawlers to analyze all sites you can reach by following links on a particular website. • Testing whether the graph is bipartite or not. • Solving puzzle games

Uniform-Cost-Search **(UCS)**	Expand the lowest cost node. The algorithm never expands a node with a cost > cost of the shortest path in the graph.  source: UCS - educba	● Solving puzzles with only one solution, such as mazes. ● Find the shortest path when the weight of each edge between any two nodes is NOT the same.
Greedy Best-First Search (GBFS)	Expand the closest node to the goal by evaluating each node using the heuristic function f(n) = h(n) It works very similarly to the uniform-cost search.	● Games. ● Web crawlers. In most cases: ● The performance can be improved largely by using a well-designed heuristic function. ● The time complexity can be better than BFS's.
A* search	It has combined features of UCS and greedy best-first search by combining g(n), the cost to reach the node, and h(n), the cost to get from the node to the goal: f(n) = g(n) + h(n)	● Traffic navigation system [3]. ● Games. ● Finding the shortest path. ● Real-time path re-planning of an unmanned surface vehicle avoiding underwater obstacles [4].

References

[1] Artificial Intelligence - A Modern Approach (3rd Edition).pdf

[2] Patel, Ronit & Pathak, Maharshi. (2018). Comparative Analysis of Search Algorithms.

[3]LIU Jingang, and LIU Yujun, Application of A* algorithm in Traffic Navigational System, Information Engineering and Electronic Commerce (IEEC), 2010 2nd International Symposium on. IEEE, 2010.

[4] Phanthong, Thanapong, et al. Application of A* algorithm for real-time path re-planning of an unmanned surface vehicle avoiding underwater obstacles. Journal of Marine Science and Application 13.1 (2014): 105-116.