Search Algorithms Report

# Depth-First:

**Depth-first search is used in:**

1. Topological sorting
2. Scheduling problems
3. Cycle detection in graphs
4. Solving puzzles with only one solution
5. Analyzing networks
6. Used as a subroutine in network flow algorithms such as the Ford-Fulkerson algorithm
7. Used as a subroutine in matching algorithms in graph theory such as the Hopcroft–Karp algorithm
8. Finding spanning trees

**Advantages:**

* Consumes less memory
* Finds the larger distant element (from source vertex) in less time.

**Disadvantages:**

* May not find optimal solution to the problem.
* May get trapped in searching useless path.

# Breadth-First:

**Breadth-first search is used in:**

1. Shortest Path and Minimum Spanning Tree for unweighted graph
2. Peer to Peer Networks
3. Crawlers in Search Engines
4. Social Networking Websites
5. Broadcasting in Network
6. It also shares uses with depth first like cycle detection, implementing Ford-Fulkerson algorithm, path finding, etc.

**Advantages:**

* Always finds the best solution

**Disadvantages:**

* All of the connected vertices must be stored in memory. So consumes more memory

# Best-First:

**Best-Fit search is used in:**

1. Web Crawlers
2. Games

**Advantages:**

* It is more efficient than that of BFS and DFS
* Time complexity of Best first search is much less than Breadth first search.
* The Best first search allows us to switch between paths by gaining the benefits of both breadth first and depth first search.

**Disadvantages:**

* Chances of getting stuck in a loop are higher.

# Uniform Cost:

**Uniform Cost search is used in:**

1. Artificial Intelligence
2. Games

**Advantages:**

* Uniform cost search is optimal because at every state the path with the least cost is chosen.

**Disadvantages:**

* It does not care about the number of steps involve in searching and only concerned about path cost. Due to which this algorithm may be stuck in an infinite loop.

# A\*:

**A\* search is used in:**

1. Pathfinding
2. Graph traversal
3. Games
4. Web-Based Maps

**Advantages:**

* It is complete and optimal.
* It is the best one from other techniques. It is used to solve very complex problems.
* It is optimally efficient, i.e. there is no other optimal algorithm guaranteed to expand fewer nodes than A\*.

**Disadvantages:**

* A\* Search Algorithm doesn’t produce the shortest path always, as it relies heavily on heuristics / approximations to calculate them.
* This algorithm is complete if the branching factor is finite and every action has fixed cost.
* The speed execution of A\* search is highly dependent on the accuracy of the heuristic algorithm that is used to compute h (n).