Abhishek Kumar Singh
MTECH (CSE) 24-26
Roll No. – 2402res24
Adm. No. – IITP002303

Artificial Intelligence
Assignment–I

Assignment – (1)

Implement BFS and DFS on a 3×3 Grid or Implement 8 Puzzle Problem.

Let take Intial State of Grid,

Intial State	→	Target State
6 3 7		1 2 3
5 1 4		4 5 6
8 2 B		7 8 B

For 20 different observations, where each intial state was different, it is observed that for majority of the case BFS is reachable with an average of 24 steps and average time take 6.538 seconds.

But for all the cases DFS was not reachable. By DFS method goal was unattainable, although taking significantly less time avg. time 3.38 seconds.

Conclusions:

BFS

→ shows consistent performance in finding solutions within a reasonable time frame.

→ was able to find solutions in a range of steps and times, demonstrating versatility

→ For some cases, not able to find sdutions for some intial cases/states within the time limit, indicating potential limitations in handling complex states.

DFS

→ struggles to find sdutions within a reasonable time frame

→ mostly unable to find solutions suggesting inefficiency in exploration or handling large states.

→ shows variability in time taken for different observations., indicating Pontential sensitivity to intial conditions.

⇒ BFS Breadth First Search

BFS is faster in finding solutions when the depth of the solution is relatively shallow or when closer to the root node.

Example.

BFS successfully found solutions within a reasonable time frame for various states with varying depths. But in observation 2 BFS found solution in just 0.799 seconds when intial state was such:

Intial State

2	8	7
1	5	3
4	6	B

Target state

1	2	3
4	5	6
7	8	B

It is suitable for shorter paths, as it provides Optimal solution in terms of number of steps.

DFS (Depth - First Search)

DFS tends to be slower in finding solutions, especially when the solution is located deep within the search space.

Example:

DFS was unable to find solution in observation 9th when within 3.65 seconds

Intial State

3	7	6
8	5	4
2	1	B

Target state

1	2	3
4	5	6
7	8	B

DFS may / might be suitable for cases / states where the search space is very large. and memory constraints are concerns as it consumes very less memory space.

→ Overall, BFS is generally faster when the solution is closer to the root node or when the shortest path is desired. while DFS may struggle in states / cases with deep search spaces.

However, suitability of each ~~dfs~~ algorithm depends on the specific problems and search spaces.