AI Assignment - 1

Question - 1 to 4 Applying BFS, DFS, UCS and A* algorithm

In this problem, we have to implement Breadth-First Search (BFS), Depth-First Search (DFS), Uniform-Cost Search (UCS) and A* algorithm to find the way for Jon Snow to reach the target, Cersei Lannister. I implemented these algorithms and tested it for 5 different maze sizes (8X8, 16X16, 32X32, 64X64 and 128X128). In the following table, I have shown the comparison between different algorithms for different maze sizes

Size of the Maze	Algorithm Used	Total Time Taken (milliseconds)	Memory Used by Stack	Total Nodes Expanded	Total Cost
8 X 8	Depth First Search	0.3	4	21	26
	Breadth-First Search	0.48	4	31	26
	Uniform Cost Search	0.8	4	30	18
	A* algorithm	0.51	4	27	18
16 X 16	Depth First Search	1.55	6	101	58
	Breadth-First Search	4.03	7	121	58
	Uniform Cost Search	2.3	10	122	42
	A* algorithm	1.41	12	93	42
32 X 32	Depth First Search	3.55	13	429	122
	Breadth-First Search	4.26	18	491	122
	Uniform Cost Search	10.5	15	487	90
	A* algorithm	5.7	19	372	90
64 X 64	Depth First Search	7.89	30	913	250
	Breadth-First Search	21.98	36	2044	250
	Uniform Cost Search	33.04	37	2012	186
	A* algorithm	31.14	53	1740	186
128 X 128	Depth First Search	18.02	61	2103	562
	Breadth-First Search	77.84	70	8181	562
	Uniform Cost Search	123.01	66	8182	430
	A* algorithm	124.38	88	7841	430

- In this memory means the maximum number of nodes in the Stack / Queue / Priority Queue at any point of time.
- The Time taken calculation is for the Algorithm to execute not for the whole program to run. The parameters calculated above are for running the same maze together. (I ran all the algorithms and then compared the results)
- Total cost has been calculated as said in the assignment.

Observations

1. From the above table, we can notice that the cost needed to reach the final goal is minimum for Uniform Cost Search and A* Algorithm. It happens because of the optimality property of these searches. We can also notice that the cost needed to reach the goal for BFS and DFS is also the same, it happens because the path to reach the goal state is only one and hence DFS always finds the most optimal path. The difference between the cost of A* algorithm and BFS is observed because the cost moving downwards for BFS in 2 whereas for A* algorithm it's 1.
2. We observe that mostly the number of nodes expanded for the A* algorithm and the DFS algorithm is the least whereas for the BFS algorithm we need to expand most of the nodes because of the fact BFS works level-wise because of which it has to expand almost all the nodes before reaching the goal. For the bigger graphs, A* Algorithm has to expand almost all nodes because the goal is at the bottom right corner and start position is at top left, so in every direction, the heuristic decreases and does not provide any benefit to the search.
3. For the small graphs, A* algorithm expands less nodes but takes time equivalent to the DFS because of the fact A* algorithm also have to calculate the heuristic function because of which it generally takes a little more time than the UFS, BFS, and DFS.
4. The memory taken for storing the nodes in the frontier is maximum for A* algorithm and BFS algorithm. This happened because the Heuristic function for A* algorithm was underestimating the goal by a huge margin for every node and hence we explored most of the nodes. Next, for BFS it is large because it works in a level-wise manner and it has to store the nodes of at least 2 levels.

Question - 5 Working with the new environment

In this case, I have modified the previous environment and made it a multiple target environment. Now in this environment, we can have 2 goals (out of which 1 will be always reachable). In this environment, we can appreciate the use of the A* algorithm as it can reach the goal state much faster as it is picking up the optimal goal state.
In this state, I observed that DFS was not able to reach the optimal answer due to which it was expanding many nodes and exploring many new ways and hence, wasting a lot of time. Whereas, in contrast, the A* Algorithm found the optimal answer and hence performs much faster.