Artificial Intelligence (CS-537)
PROJECT-1 The Searching Pacman Problem

Submitted By -
Dushyant Goyal (109323567)
Bhavya Agarwal (109335580)

Profing Tool - python memory_profiler

1. Depth-first search (DFS)

Maze	SearchAgent	Heuristic	No. of Nodes Expanded	Running Time (sec)	Memory Usage (MB)
tinyMaze	-	null	14	0.0	0.004
mediumMaze	-	null	144	0.0	0.340
bigMaze	-	null	390	0.0	0.840

2. Breadth-first search (BFS)

Maze	SearchAgent	Heuristic	No. of Nodes Expanded	Running Time	Memory Usage
tinyMaze	-	null	15	0.0 sec	0.004
mediumMaze	-	null	267	0.5 sec	0.496
bigMaze	-	null	617	1.1 sec	1.168

3. Uniform-cost search (UCS)

Maze	SearchAgent	Heuristic	No. of Nodes Expanded	Running Time	Memory Usage
mediumMaze	-	null	271	0.0	0.590
mediumDottedMaze	stayEast	null	187	0.0	0.500
mediumScaryMaze	stayWest	null	109	0.0	0.312
bigMaze	-	null	623	0.1	1.316

4. A* graph search

Maze	SearchAgent	Heuristic	No. of Nodes Expanded	Running Time	Memory Usage
bigMaze	-	manhattan	550	0.1	1.161

5. Corners Search Problem

Maze	SearchAgent	Heuristic	No. of Nodes Expanded	Running Time	Memory Usage
tinyCorners	bfs	null	243	0.1	1.161
mediumCorners	bfs	null	1921	0.2	3.423

6. Corners Heuristics

Maze	SearchAgent	Heuristic	No. of Nodes Expanded	Running Time	Memory Usage
mediumCorners	A* cornersAgent	min distance from corners	1527	0.1	2.712
mediumCorners	A* cornerAgent	max distance from corners	1140	0.2	2.023

7. FoodSearchProblem (Eating all the dots)

Maze	SearchAgent	Heuristic	No. of Nodes Expanded	Running Time	Memory Usage
trickySearch	A* searchAgent	min distance from corners	17208	7.5	30.53
trickySearch	A* searchAgent	max distance from corners	10101	4.4	17.21
trickySearch	A* searchAgent	Minimum Spanning Tree (MST)	7116	3.4	12.62

The above table summarizes the result of different search algorithms.
Here is our observations-

- BFS will look at more nodes than DFS, However, DFS is faster.
- The solution from BFS will be more optimal than DFS

- A* with a good heuristic expands fewer nodes than BFS.
    - Manhattan heuristic- 550 nodes expanded

- Corners Heuristics
    - Minimum distance from corners - Minimum distance of all corners to the current node (because we wish to reach that goal fast)
    - Maximum distance from corners - Maximum distance of all corners to the current node (because we need to cover at least that maximum distance)

  These heuristics are both consistent and admissible. We are not overestimating the actual cost. The actual cost will always be greater or equal to the maximum of the cost. Also, the step cost of maximum using manhattan distance is not greater than the actual total cost.

- Food Search heuristic
    - Minimum distance from corners - Minimum distance to all food items to the current position
    - Maximum distance from corners - Maximum distance among all remaining food items
    - Minimum spanning tree - For the current position of the node. We start from the nearest food the node has. After that we keep recursively going to the next nearest node until we exhaust the list of food nodes. It expands close to 7100 nodes
    - The property of going to the next nearest node in itself is admissible and consistent since, it gives us the minimum cost of the path to all the goal items which is less than the actual cost.