REPORT

**Goal**

This project is aimed at designing an agent that is able to find optimal paths through its maze world which intends to reach particular locations and eating all the dots in minimal steps. We have used multiple search strategies both informed and informed search to achieve this goal. Uninformed search strategies involved BFS (Breadth First Search), DFS (Depth First Search), UCS (Uniform cost search). Informed search strategy included A\* search with proper heuristics to get the optimal results.

**Learning Curve**

We had good progress in gaining insights regarding various search strategies. We came across the advantages, optimality, completeness, space and time complexity of multiple search strategies. Some of them are as follows :

1.DFS and BFS comparison :

We compare the efficiency and complexity of the two solutions using :the solution length which means the number of edges in the solution, and the number of nodes expanded to get to the solution.

For DFS, solution length is 130 and the nodes expanded are 146 whereas for BFS these are 68 and 269 respectively.

So we can infer that DFS is a very space efficient algorithm as it only expands 146 nodes for a 130 edges long solution as compared to BFS which expands 269 nodes for a 68 edges long solution.

But, BFS is very optimal as compared to DFS as inferred from the respective solution lengths but it will be inefficient where number of nodes is very large

3. Uniform Cost Search is for weighted graphs and is similar to BFS. It expands the frontier only in the direction which will require the minimum cost to travel from initial point among all possible expansions. So , it is a more general form of BFS , saying that BFS is a case of UCS when expansion cost is uniform all over the graph.

4. A\* search finds the shortest path through a search space to goal state using heuristic function. The heuristic function is represented by the equation f(n) = g(n) + h(n) where g(n) is the distance from source to node n and h(n) is a heuristic function to estimate the distance from node n to goal state. Think of g(n) & h(n) as two forces in opposite directions. If search goes way too deep in a path where the heuristic function doesn’t have much to promise, then g(n) pulls it back to relax more promising paths.

**Challenges**

Implementing non-trivial heuristic for the A\* search to find the corners of the map was one of the most challenging tasks since it involved coming up with an admissible and consistent heuristic. Finding all the corners using A\* search is a comparatively easy task but when we need to implement the same using heuristics, it becomes a different ball game. We actually spent quite a bit of time for coming up with a proper heuristic to solve this problem. Our initial solution was based on the assumption that the agent walks to the closest unvisited corner and once the agent has already reached the corner, then it only needs to walk along the borders of the map to reach all the unvisited corners. So we only need to calculate the Manhattan distances between current position and unvisited corners to find the minimum distance. But unfortunately, we didn’t go for this approach. We went for a different approach in our implementation where we ignored all the walls in the map, hence the cost of walking from one point to another is the Manhattan distance between these two points. For all corners unvisited, the corner with the largest Manhattan distance is the last one to be visited. Once we reach this corner, the game reaches its goal state.

**Accomplishment** :

Successfully implemented and learned all the above stated search strategies. It was very interesting to implement each and every search and we also learnt the importance and optimality of each search algorithm. How and where these are relevant. We learnt to make use of state knowledge to make our search smarter using informed search strategies. We also learnt the importance of heuristics and difficulty faced while choosing a heuristic as will be discussed in the challenges section. How the choice of a heuristic is at the heart and core of an A\* search algorithm. It was very exciting to use the theoretical knowledge gained in class and make our pacman smarter.