COMP 4106 Assignment One

Connor Stewart (101041125)
Gabrielle Latreille (101073284)
Abdaljabbar Hersi (100919600)

1 – Abstract

This report provides information on a self-directed project on informed graph searches with the A* search method. The project is to design and implement the well-known A* search algorithm with a custom-designed agent interface and heuristic model. Input for the program is in the form of comma-separated value (CSV) files. Designated files store the output of the algorithm, and results are also written to the command line. Further in the report herein, the functioning of the program compared to other models is evaluated. Discussions on the methods used will also be touched upon in greater detail.

2 – Introduction

The assignment's primary task is to analyze the CSV datasets to determine the shortest path costs using a custom A* search algorithm. This algorithm is to work out the minimum cost path between the starting and goal nodes. A custom heuristic was implemented in the A* program for the input graph-matrix; these heuristics relied on the minimum Manhattan distance between the nodes as estimates for minimum paths. The Manhattan distance is an approach to measuring the distance that relies on taking the distance between two points measured along axes at right angles.

3 – Datasets

Input for the program comes from comma-separated value (CSV) files, with the data being organized into a grid to describe the environment. The grid in the input files has one S and one or more G's to represent the respective starting and goal nodes. The first index in the grid indicates the row, and the second the column, and both rows and columns are zero-indexed. The program will output to the first text file the optimal path from the starting to goal nodes; this is formatted as a list of then positions along the optimal path from the starting to the goal state. The program writes to a second text file the ordered list of nodes explored during the search. The dataset contains a grid representation of a set of paths, with a starting node that the search starts from and one or more goal nodes that the searches end on. The input-grid also contains X's, representing walls that the path cannot traverse, and positive real-values indicating the movement cost to move onto a tile.

4 – Results and Answers

The implemented algorithm correctly outputs pathing information for both examples one and two in the provided assignment. The path costs, paths, and explored nodes match the solutions provided.

Below is a list of answers to the provided questions in the assignment.

1. *What type of agent have you implemented:*

We have implemented a goal-based agent, and the goal is to minimize the cost of getting to the goal node.

2. *What is the best heuristic to use in A* search for this environment? Recall that there may be multiple goal states:*

   The best heuristic to use is the Manhattan distance between a node and a goal. When there are multiple-goal states, the minimum value between the node and each goal state can be used to minimize the heuristic. The Manhattan distance is calculated by adding the distance in the x-axis to the distance on the y-axis.

3. Suggest a particular instance of this problem where A* search would find the optimal solution faster than uniform cost search:

S	(A,1)	(B,3)	X
(C,2)	X	(D,2)	G
(E,3)	X	X	(F,1)
(H,2)	(I,1)	(J,1)	G

Uniform Cost Search: Using Example 2 - (n-pathcost, parent)

Step 0: Frontier - S (0,None)   Explored:

Step 1: Frontier - A (1, S)   Explored: S (0,None)

- C (2,S)

Step 2: Frontier - C (2, S)   Explored: S (0,None), A (1, S)

- B (4,A)

Step 3: Frontier - B (4, A)   Explored: S (0,None), A (1, S), C (2, S)

- E (5,C)

Step 4: Frontier - E (5, C)   Explored: S (0,None), A (1, S), C (2, S), B (4, A)

- D (6,B)

Step 5: Frontier - D (6, B)   Explored: S (0,None), A (1, S), C (2, S)

- H (7,E)   B (4, A), E (5, C)

Step 6: Frontier - G (6, D)   Explored: S (0,None), A (1, S), C (2, S),

- H (7,E)   B (4, A), E (5, C),D(6,B)

Step 7: Goal State is reached

As can be seen above uniform cost search takes about 7 steps, and while checking the output of our A* search we concluded that it took about 6 steps.

4. Suggest a particular instance of this problem where a greedy heuristic search would not find the optimal solution:

(S,4)	(A,3)	(B,2)
(C,3)	X	(D,1)
(E,2)	(F,1)	(G,0)

Greedy Heuristic Search: Using Example 1 - (Heuristic, parent)

Step 0: Frontier - S (4,None)   Explored:

Step 1: Frontier - A (3,S)   Explored: S (4,None),

- C (3,S)

Step 2: Frontier - B (2,A)   Explored: S (4,None), A (3,S),

- C (3,S)

Step 3: Frontier - D (1,B)   Explored: S (4,None), A (3,S), B (2,A)

- C (3,S)

Step 4: Frontier - G (1,D)   Explored: S (4,None), A (3,S), B (2,A) , D (1,B)

- C (3,S)

Step 5: The Goal state has been reached

The above example shows a case where the greedy heuristic does not find the optimal solution we know the because the path cost of S->A->B->D->G is 8 while the path cost of S->C->E->F->G is 7.

5. *Consider a variant of this problem where the agent can move to adjacent diagonal squares. What would be the best heuristic to use in A* search if the agent could also make such diagonal moves:*

   The best heuristic for this problem would be the Euclidean distance heuristic. The Euclidean distance can be calculated using the Pythagorean theorem, i.e., the shortest distance from the node to the goal.

   $\sqrt{(distance\ on\ x-axis)^2+(distance\ on\ y-axis)^2}$.

5 – Discussion and Conclusion

In conclusion, the algorithm correctly models the problem and produced results in a good timeframe. The heuristics used well represent the environment and help reduce runtime.

6 – Statements and Contributions

Overall, the project was completed in group meetings through discord. All group members made significant contributions to the overall project, with approximately equal contribution among each member. Since the group work was evenly divided amongst members and completed in a shared environment, all members of the group completed all aspects of the assignment together.