REPORT

a-) Explain how you decide on the heuristics you determine.

- **In A* Search using admissible heuristics**, I decided to use euclidean distance as an admissible heuristic. A heuristic is admissible if it does not overestimates the cost of reaching from any tile to goal tile. The straight-line from any point to goal is always lower than other paths according to triangle inequality, and for that reason I used Euclidean distance as an admissible heuristic for any tile.
- **In A* Search using inadmissible heuristics**, I used that strategy: First I performed A* Search with admissible heuristics, and when I found the optimal solution with that search, I tried to change three or four of the admissible heuristics, and I have made them inadmissible multiplying their euclidean distance with some costants and that made them to overestimate the cost of to reach from these tiles to goal tile. And because of that over estimation, the agile choosed wrong paths according to that wrong, inadmissible heuristic, so at the and the optimal solution could not be found.

b.) Comment on your findings with each approach (a, b, c-i, c-ii, d-i and d-ii):

***** Uniform Cost Search(without extra move points) = Breadth First Search**

This algorithm has the same procedure with Breadth First Search if additional move points are not taking into consideration, and answers to questions in BFS is the same with that approach. So, I did not add the answers for that approach below.

1-) Can the agent find all possible paths from the starting tile to the goal tile? How can this be made possible?

- **Breadth First Search:**

  In that algorithm, the agent first expands the root node, then all nodes in level n, then all nodes in level n+1 and so on. Whenever it finds the goal node, it finds the optimal path and returns it and discards all nonoptimal paths. Suppose that goal node is in level n, and BFS finds all nodes up to level n and also it finds all paths from starting tile to that level which includes goal tile. Because it checks all paths equally from starting tile to goal tile, it finds all possible paths from starting tile to goal tile. However at the end, just the optimal one of the paths is returned.

- **Depth First Search:**

  In that algorithm, the agent chooses a path to itself, and follows that path until it ends. If goal is not found, it goes back and continue with shallowest node with the same operation. So it cannot find all possible paths from starting tile to goal tile.

- **Uniform Cost Search(with extra move points)**

  In that algorithm, procedure is similar with Breadth First Search, but this time the next node is chosen from frontier with minimum path cost. It cannot finds all possible paths from starting tile to goal tile every time. If we fix extra move points such as UCS should check all nodes from starting tile to goal tile, it finds all possible paths. And also if all extra move points are the same, UCS is identical to BFS.

- **A* Search(with admissible heuristics)**

  In that approach, the only difference with Uniform Cost Search is heuristics which I determined. It adds that heuristic to path cost in any node and make decision according to that addition, chooses the smallest one. For that reason, it cannot find all possible paths from starting tile to goal tile every time. However, if we fix heuristics and extra move points such as A* should check all nodes from starting tile to goal tile, it finds all possible paths.

- **A* Search(with inadmissible heuristics)**

  In that approach, heuristics directs agent to wrong way and the agent cannot find optimal path because of that wrong path estimates. All possible paths cannot be found with that approach. If that wrong estimates force agent to check all possible paths from starting to goal tile, that approach can find the all possible paths.

2-) Can the agent find the path from the starting tile to the goal tile with minimum number of actions? If so, how? If not, what are the causes?

- **Breadth First Search:**

  In that approach, the agent expands all nodes equally until the goal node, so it cannot find the path with minimum number of actions, it always should check all nodes in level n, level n+1 and so on until goal node.

- **Depth First Search:**

  DFS first chooses a path to itself, and if the goal node is in that path, it finds the solution quickly. In our case, once the agent come to node in coordinat (0,2), if it chooses itself the vertical path from (0,2) to (4,2) = goal node, it can find the path with minimum number of actions.

- **Uniform Cost Search(with extra move points)**

  UCS also cannot find the path with minimum number actions, because in every step, the path cost is increasing and the agent should expand the lower cost node in every step. In our case, when the agent is in (0,2), it cannot directly go to (4,2) because path cost is increasing every time when we go to down.

- **A* Search(with admissible heuristics)**

  Admissible heuristic never overestimates the path from starting tile to goal tile, and the agent will reach the goal tile with lower number of actions than Uniform Cost Search, because at that time heuristics are also important for choices of the agent and agent will go towards goal more quickly. But because of path cost is increasing with extra move points every time, the agent cannot reach to goal node with minimum number of actions.

- **A* Search(with inadmissible heuristics)**

  In that approach, If I determine heuristics of nodes in (0,3)(1,3),(2,3),(4,3) very high such as the agent do not chooses these nodes from (0,2) to (4,2)=Goal State, the agent can reach to goal node with minimum number of actions

3-) Can the agent find the path that it reaches the goal tile with the minimum cost in movement points? If so, how? If not, what are the causes?

- **Breadth First Search:**

  In that approach, if all move points are the same such as in our case, the agent can reach to goal node with minimum cost in move points. Because, the agent equally looks for all nodes in level n, and level n+1 and so on. Whenever it finds the goal tile, it returns the path and if all move points are the same, that path will be the optimal one.

- **Depth First Search:**

  DFS first chooses a path to itself, and if the goal node is in that path, it finds the solution quickly. In our case, when the agent is in node (0,2) if it chooses the node (1,2), it can reach to goal tile(4,2) with minimum cost in move points, however, if it chooses the node (0,3), it cannot reach to goal tile with minimum cost in move points because DFS follows that path until it ends.

- **Uniform Cost Search(with extra move points)**

  UCS also can find the path with minimum cost in move points, because it always chooses the node which has the smallest path cost from frontier. So, at the end, founded path will be the optimal one, because the agent will reach that path choosing minimum cost path in every step.

- **A* Search(with admissible heuristics)**

  In that approach, the heuristics do not direct the agent to wrong path which is not optimal, so that approach also finds the path with minimum cost in move points.

- **A* Search(with inadmissible heuristics)**

  In that approach, heuristics can direct the agent to the path which is not optimal, and because of that, that approach cannot finds the path with minimum cost in move points.

4-) What would happen if additional movement points were negative?

- **Breadth First Search:**

  Suppose the agent is in (0,0), and the tile (0,1) has extra move points 5, to expand from (0,0) to (0,1) has path length 1, however has path cost 5 + 1 = 6.
  BFS finds the optimal path from starting tile to goal tile according to path length, not to path cost which comes from move points. So, if the move points in the state space are equal, the founded optimal path according to path length will be optimal. However if the move points on that path is negative, that can make that optimal path non-optimal, but choice of the agent cannot be affected from these negative move points, because it consider the path length to make a choice. Completeness of the BFS is not affected from these negative move points.

- **Depth First Search:**

  Depth first search also makes its decision according to path length. So DFS chooses its path with same procedure if some move points were negative. Step costs on that chosen path can make that chosen path optimal or non-optimal, but choice of the agent cannot be affected from these negative move points. Completeness of the DFS is not affected from these negative move points.

- **Uniform Cost Search(with extra move points)**

  In that case, negative move points affects the choice of the agent. Uniform Cost does not take care of path length, it just take care of the path costs, and if some move points are negative, that cannot affect the optimality of the Uniform Cost Search, because the agent always chooses the minimum cost path, however these negative move points can affect the completeness of the UCS.

- **A* Search(with admissible heuristics)**

  In that approach, the affects of the negative move points are similar with UCS. The heuristics will always direct the agent to minimum cost path, and the choices of the agent will be affected from these negative move points, however optimality of the A* Search will not be affected, the agent always chooses his way with minimum of the heuristics + move points and resulting path will be optimal. However, completeness of the A* Search will be affected from these negative move points.

- **A* Search(with inadmissible heuristics)**

  In that approach, wrong heuristics and also these negative move points can make the search incomplete and non-optimal.

Hasan YENİADA

220201024