Project 01: The Searchin' Pac-Man

Pratik Sushil Zambani, Swetha Tatavarthy

1.Depth First Search

Datastructures used and information passed to it:
Stack - node_state, actions to reach the node (For Last In First Out Traversal)

Explored_list - nodes/states (To keep track of visited nodes)

2.Breadth First Search

Datastructures used and information passed to it:
Queue - node_state and actions to reach node (For First In First Out Traversal)

Explored_list - nodes/states (To keep track of visited nodes)

3.Uniform Cost Search

Datastructures used and information passed to it:
Priority Queue - node_state, actions to reach node, cost to reach node
priority = cost to reach node

Explored_list - nodes/states (To keep track of visited nodes)

4.Astar search

Datastructures used and information passed to it:
Priority Queue - node_state, actions to reach node, cost to reach node
priority = cost to reach node + heuristic value

Explored_list - nodes/states (To keep track of visited nodes)

5. Corners problem

Encoding for state:
Encoding used for state here is the state position and the set of visited corners when the state if encountered. I.e (state, set(visited_corners))

Reason: Every time a successor is encountered we check if that state is a corner and add it to the set of visited corners reached till that state. This set can then be used for verification of goal state. Also usage of 'Set' guarantees uniqueness and eliminates the need for having to check for duplicate insertions.

6. Corners Heuristic

The heuristic used for corners problem calculates the nearest corner from the start state using the manhattan distance. Then considers this state as the start state and computes the distance to the next nearest corner. It continues to do this until all corners are covered adding the distance each time and returns total distance. The heuristic is admissible because it computes the manhattan distance from the start state to nearest corner and so on without accounting for the walls. Also there is a possibility that a path exists that does not traverse the nearest corner first. In both these cases the value of the heuristic is an underestimate of the actual cost. The heuristic is consistent because a state that is closer to a corner than a previous state will return a smaller heuristic value which will be the case in the actual path as well.

7.Food Heuristic

We calculate the manhattan distance to all the goals from the start state and take the maximum of these values . Then we calculate manhattan distance from the goal state of previous point to all other goals and take minimum value. Heuristic returns sum of these two values. At any given state, the heuristic will consider the maximum farthest goal and the nearest one from that point. This balances the heuristic cost and will be lower bound on the actual cost, since majority part of heuristic cost is considering only farthest goal. As we move closer, the first part will give reduced cost resulting in a monotonically decreasing value.