Corners Problem – Q5 & 6

1. The Pacman corners problem can be described as solving the problem of finding a path that passes through a set of specific points, in this case the corners of a maze. The most important aspects of the problem are the initial game state, the corner positions, what a state holds, and the goal requirements.
   * Initial game state: Holds all the essential information for a game. Pacman’s starting position, the position of the corners/goals, and position of walls. To solve the problem we need to use the first two.
   * Corner positions: in our case the coordinates were: ((1,1), (1,top), (right, 1), (right, top)) which are the 4 corners of the grid. These are important because they give us the locations Pacman has to pass through to win the game.
   * State composition: A state for this problem is made up of two aspects, Pacman’s current position, and a Boolean list indicating which corners have been visited.
   * Goal state: the goal state is when Pacman has visited every corner coordinate. This is represented by a list of True values.
2. The heuristic for this problem is a sum of the manhattan distances between the relevant points. We start off by finding the corner with the shortest manhattan distance between the starting position and itself. We add this distance to the heuristic cost and update our position to the closest corner found. Then we iterate through the list of unvisited corners, always finding the shortest distance between the current position and the remaining corners, and updating the current position to the closest corner, and adding that distance to the heuristic. We do this until there are no more corners unvisited. At the end our heuristic cost will be the sum of each shortest manhattan distance between corners, and the initial shortest distance between the starting point and the corner.
3. To show the heuristic is consistent we must first show it is admissible:
   * Admissibility: We can say that the cost of this heuristic will never be higher than the actual cost. This is because the manhattan distance measures the displacement distance between two points, and in this case it does not take obstructions such as walls into account. Pacman can only move in four directions, and not diagonally so in most cases Pacman’s path will always be longer than the manhattan distance. Even in the case where the manhattan distance is a straight line in a direction Pacman can move, it will at most be equal to Pacman’s actual path. The fact that the manhattan distance does not take obstacles into account also means that it will be shorter than the actual cost, since Pacman has to move around walls, adding extra cost to the path. This heuristic remains admissible even taking the sum of manhattan distances, as the explanation above remains constant between each corner. This is derived from the problem of finding the shortest distance travelled between all goal points (corners).
   * Consistency: This heuristic will not drop more than the cost of the action chosen. Since it is admissible, we know the heuristic cost of paths will be lower than the actual cost. When Pacman takes an action, in this case a path to reach a goal, the heuristic will drop at most the shortest manhattan distance to that goal as the cost is calculated from the current position to the goal, when that goal is reached, that cost is dropped from the heuristic. This is why when all four corners are visited we have a heuristic cost of zero, as it has dropped manhattan distance cost of getting to each goal.

Eating All the Dots Problem – Q7

1. The eating all the dots problem can be described as finding the shortest path which passes through every specified point.
   * Initial game state: Holds all the essential information for a game. Pacman’s starting position, the position of the food dots, and position of walls. To solve the problem we need to use the first two.
   * Dots positions: A 2D Boolean list or a list of coordinates specifying where each food dot is placed. Important as we need to know the position of dots in relation to Pacman.
   * State composition: The state is made up of Pacman’s position, and a grid of food positions.
   * Goal state: Reached when all the food on the grid are eaten. When the whole list is false.
2. The heuristic for this problem is derived from finding the furthest distance to a food dot. We have take Pacman’s current position and a list of the food coordinates. We then iterate through all the food in the list of coordinates to find the one with the furthest maze distance from Pacman’s current position (Maze distance is the distance Pacman must travel to a specified position taking walls into account. Given to us in the code.). The heuristic cost then becomes the furthest distance to Pacman’s current position.
3. To show the heuristic is consistent we must first show it is admissible:
   * Admissibility: This heuristic is consistent as Pacman’s minimum path cost will be the cost of travelling to the furthest food in the grid in the situations where there is only one food or all foods are eaten on the direct path to the furthest food. So as Pacman moves across the grid the furthest food might change, but the heuristic cost won’t go above the actual cost since Pacman will have to travel to the furthest food in the board at some point, adding that cost to the actual cost.
   * Consistency: This heuristic is consistent since the distance between the food stays constant the variable that changes is Pacman’s position, but if Pacman chooses a certain action, eating a specific food, the heuristic cost will change to the next furthest food, but since the distance between food is constant in the maze distance, the path cost to the next furthest fruit will not be higher than the cost of the path Pacman just took. This continues until Pacman eats all the dots, and the heuristic cost returns to zero as he reaches the last food in the grid, and that becomes his position.