Ege Dinçer
60190

Q1:

BFS is always able to find the shallowest solution but DFS doesn't guarantee this. DFS is also not optimal but BFS is if the costs are equal. However, BFS requires more expansion of nodes than DFS. I would prefer BFS if I want to find the shallowest optimal solution because DFS has a high probability to find a non-optimal solution. If I have a limited space and the problem-tree is very deep I would prefer to use DFS as well as if I'm expecting solution to the problem at the deepest levels, I will use DFS because it will use lesser space and would be faster than BFS.

Q2:

In UCS we explore every direction but in A* search we expand more toward our goal with an optimal/admissible heuristic. A* search minimizes the number of expanded nodes compared to UCS. Therefore, the path cost and the number of expanded nodes will be lower than the UCS. If I want a quick optimal solution and I have a good admissible and consistent heuristic I would choose A* over UCS since it will find the optimal solution very quickly. However, if I don't have and good heuristic and I want to find the optimal solution, I will use UCS instead of A* since it will find optimal solution with the optimal cost.

Q3:

I have retrieved the corners as a list and initial coordinates as my starting state after that I append my successors of the state to a list with the successor method with adding a cost of 1 to each of successors. I remove the corners from the state as the Pacman reaches to corners to make it not visit the corner again. I also take care of the walls as to make Pacman not hit the walls. With this implementation it allows me to find the corners (Not optimally by the way) and remove the visited corners as the corners are my goal states.

Q4:

I have implemented the Manhattan distance as my heuristic to find the corners in an optimal way because Manhattan distance allows me to find a shorter path to reach to the goal. I haven't used Euclidean distance since Pacman is unable to move diagonally. Manhattan distance is admissible since it doesn't overestimate the number of expansion of states to reach to the goal since the Pacman can only move 1 step each time as we're getting

rid of walls in the successor method, we have an unobstructed path. Since we are constrained to move either horizontally or vertically this heuristic is consistent.

Q5:

I have used the BFS for eating all the dots problem since BFS allows me to find the shortest, most optimal solution which is the closest food in the grid that are mostly near to the Pacman. It allows to find the closest dots without needing to expand to the deepest node. It is admissible because it guarantees to find the optimal solution if there exists any. BFS expands to the next depth after exploring all the nodes in the current depth. With this it guarantees to find the shortest optimal solution. Since our moves and expansion nodes are constrained BFS heuristic is consistent.

Q6:

Inadmissible heuristic overestimates the solution which results in a non-optimal cost with a larger number of expanded nodes. Consistent heuristic estimates the solution either less than or equal to the neighboring goal state. Inadmissible heuristic can be used if we don't know what the optimal solution for the problem is and we're is unable to make estimations to limit the expansion of nodes. This will be costly, but it will lead to some solution. I would use a consistent heuristic if I want to find the optimal solution and find it fast with least node expansion compared to Inadmissible heuristic. However, I need to be sure that this heuristic is working, and it leads me to a solution and also, I need to know that the relationship is preserved between each node.