

Q1)

Bfs typically expands more nodes than Dfs, but Bfs always finds a shorter path. Dfs is better when the goal is deep, Bfs is better when the goal is not far and the memory is sufficient.

Q2)

A* expands less nodes than UCS, using UCS we will find the path with the lowest cost in comparison to A* which focuses on getting to the goal. Use UCS when we have no idea where the goal is, and A* when we have an idea where the goal is.

Q3)

For the goal state we simply look for the first state in which we have visited all the corners, this heuristic works since I keep track of the unvisited corners and I calculate the total distance to reach the final state.

Q4)

In calculating the heuristic I check if we reached the goal state and return a heuristic value of 0 if we did, if not we get the permutations of the unvisited corners and check the total manhattan distance of currentPosition-unvisitedCorner1-unvisitedCorner2... and we return the minimum total distance. It is consistent because the closer we get to one of the points the smaller the value of the total distance (currentPosition-unvisitedCorner1-unvisitedCorner2...).

Q5)

For eating all the points heuristic I check if we reach the goal state, where there is no food left, and return 0. If we're not in the goal state I find the 5 most spread out food points and put them in a list, then I get the permutations of that list and find the 10 least total manhattan distance between them, after that I loop over the smallest 10 total manhattan distances, then find and return the smallest total maze distance.

Q6)

1.
 - a. Consistent heuristic:
 - i. We get an optimal path, so the path will have the lowest cost.
 - ii. We expand more nodes.
 - b. Inadmissible heuristic:
 - i. We may find a suboptimal path, which will cost more than the consistent heuristic path.
 - ii. We expand less nodes.
2. Prefer consistent heuristic when the optimal solution is required.
3. Prefer inadmissible heuristic when we want a quick solution, and when we don't care if the solution is optimal.