# Comp341 Assignment 1 Report

Name Surname: Burak Yıldırım
Student ID: 72849

Q1)
In the depth first search, we expand the deepest node first and we use stack as a frontier. In the breadth first search, we expand the shallowest node first and we use queue as a frontier. In terms of path cost, they perform similarly since they don't include the cost for each edge while searching. Because of their expansion of node behaviors, BFS is good at finding short paths and DFS is good at finding longer paths with expansion of more nodes. This makes BFS optimal for finding shortest paths compared to DFS which is the usual case. I would prefer DFS for memory consuming paths and longer paths. I would prefer BFS for finding the optimal and shorter paths.

Q2)
A* and UCS searching algorithms can be used to find the optimal result. A* uses heuristic and cumulative costs while searching. However, UCS uses cumulative costs. UCS expands more nodes compared to A*. I would prefer UCS in case of an inadmissible and inconsistent heuristic. I would prefer A* for other cases for finding the optimal result by expanding a smaller number of nodes.

Q3)
I used the (startingPosition, (0,0,0,0)) as the state for the corners problem where the first element of this tuple is the starting position, and the second element of this tuple is a tuple that indicates 0 or 1 for each corner. 1 denotes for the goal in the corner. It allows me to solve the problem since I can check whether the current position is a goal state or not which leads me towards the solution.

Q4)
In the corners heuristic, I calculated the Manhattan distances between the current node and each of the corners that don't have food. Then, I stored these distances in a list and if the array has at least 1 element, I returned the maximum distance from this list as the heuristic. It's admissible because heuristic cost will always be less or equal to the actual cost to the corner. It's consistent because the difference of two heuristics is less or equal to the actual cost for each corner.

Q5)
In the food heuristic, I calculated the maze distance between the current node and the position of each food. Then, I stored these distances in a list and if the array has at least 1 element, I returned the maximum distance from this list as the heuristic. It's admissible because heuristic cost will always be less or equal to the actual cost to the food. It's consistent because getting closer to a food decreases the heuristic which makes it less or equal to the actual cost.

Q6)
By using a consistent heuristic, we can find an optimal path. Since calculating heuristic cost takes significant amount of time an inadmissible heuristic will be faster. But the drawback of it is generating a not optimal solution. I would prefer consistent heuristic if I want to get an optimal

solution but accepting that it could take a bit longer in terms of time. I would prefer inadmissible heuristic if I want a much faster solution and accept a not optimal solution.