<u>COMP-341 FALL 2022 PROJECT-1 REPORT</u>

Name – Surname: Barış Kaplan, ID: 0069054

1-) Although BFS expands more nodes than DFS (BFS => 269, DFS => 146), the total cost of path is less in BFS than DFS (BFS=> 68, DFS=>130). Hence, the optimality of BFS is higher but BFS needs more memory. Whereas BFS initially seeks for the shallowest element, DFS initially seeks the deepest node. Thus, using BFS for finding a deep target might bring some memory-related overflows (or errors/exceptions). For finding a deep target and utilizing less space, I would prefer DFS. If I do not care memory and the target is shallow, I would pick BFS.

2-) For A*, the number of expanded nodes is less than the UCS (A* => 221, UCS => 269). However, their total path costs are near. Thus, we can infer that the optimality of A* is higher (for cost > 0). Compared to UCS, A* additionally uses a heuristic in its frontier. Since A* expands less nodes, it needs less memory. As UCS is uninformed, it controls all directions while seeking a goal. When I have to control each direction, and cannot (or should not) keep a consistent heuristic, I would select UCS. When I can find the aim directly, want to save more memory and want to know the closeness of agent to goal, I would choose A*.

3-) I have kept the entire list of the nodes that can be visited (state[1]) and the current location of the agent (state[0]) in the state variable. Then, by checking whether the non-traversed node list length is 0 (i.e. all nodes are traversed), I have checked if we reached the goal. As the location matches of agent and non-traversed node increase, the total number of non-traversed nodes decreases. Thus, the agent becomes closer to the goal state (which indicates the problem solution).

4-) In my heuristic, for each agent location, I have taken the biggest manhattan distance from agent to a non-traversed near node. I choose this heuristic to utilize less space and higher dimensional data. Since the max distance to a non-traversed near node (estimation) cannot exceed sum of all costs to adjacent nodes (truth), this heuristic is admissible. Due to positive value returned by manhattan, my heuristic holds following: heu(y)+cost(t,y)>=heu(t) where:t=start,y =adjacent, heu(target)=0,cost(t,y)=1 (i.e. fits to triangle inequality where the triangle corners are agent location, adjacent location, goal location). For my heuristic; in other words, summation of the cost of the way from agent to adjacent and the predicted cost of the way from adjacent to goal is greater than or equal to the heuristic cost of the way from agent to goal. Therefore, my heuristic is consistent.

5-) In my heuristic, for each agent location, I have get the largest manhattan distance to a non-traversed adjacent food by applying an A* search to a position search problem. Since it provides lower cost and quicker search with A*, I have used this heuristic. Since the cost of max distance to a near food (estimation) cannot exceed sum of all costs to adjacent foods (truth), this heuristic is admissible. This heuristic is consistent because it fits to the triangle inequality where the corners are the locations of Pacman, adjacent, and food, and goal's heuristic is 0.

6-) Consistent heuristic usually expands less nodes. Moreover, since it can exceed the true path cost, inadmissible heuristic has higher path cost. When it is hard / unnecessary to seek a consistent heuristic and when we do not care memory, then I would choose an inadmissible heuristic. If I want to utilize less memory, obtain an optimal solution, and have less cost; then I would choose a consistent heuristic.