Comp341 HW1 Metehan Gelgi
64178

1)

DFS and BFS are the search algorithms which does not related to the cost of the edges. These algorithms have some differences while expanding nodes. Maybe cost differences related to the number of expanded nodes. While DFS expand nodes to the depth narrow, BFS expand nodes breadth narrow. DFs can find longer paths by expanding more nodes. However, BFS has better algorithm in terms of shorter path. Hence BFs is optimal but DFS is not optimal in most cases. If I want to find optimal and near state, I will use BFS. I will use DFS if I have memory issues due to deep nodes.

2)

A* and UCS algorithms are capable of finding both optimal paths. Main difference between these algorithms is used function for frontier. UCS finds cumulative path cost of frontiers then pops smallest one. However, A* star algorithm consider cumulative cost and heuristic cost to the goal state. A* expands less nodes than UCS. In final both can find optimal solution (in positive costs). I will prefer if our heuristic is not admissible and not consistent. I would use A* if I want to find optimal path by expanding less nodes.

3)

In four corners problem, I used (location, cornerList) as a state. This state contains our current location and list of unchecked corners. These elements are needed to solve corners problem. we are using location to match goal corners and pacman location. If they are matched, we reached one goal and then remove it from cornerList. That’s why these two of them are enough to solve problem.

4)

In this heuristic I am finding sum of Manhattan distances of the corners from closest one to furthest one. First find nearest corner and calculate Manhattan distance to this corner from starting position. Then remove this corner from corner list. Now our location is this corner. Then we are finding next closest corner from this corner. We keep going until there is no unchecked corners. Returns sum of Manhattan distances as heuristic. Sum of Manhattan distances are never greater than cost of optimal solution so heuristic is admissible. difference of heuristic of two position is equal to the Manhattan distance between these two positions. That’s why heuristic is consistent if this method is admissible.

5)

Choice of heuristic in the eating all the dots problem is finding max maze cost of foods from current location. If there is no food our heuristic is 0. Otherwise, we are getting food locations as a list then iterate over it. Finds maze distances between food locations and current location (heuristic of current location and food location). Then gets max of it. I am getting maximum of heuristics because these heuristics are admissible heuristics since we are calculating maze distance (which is no more than real distance) and max of admissible heuristics is admissible. This heuristic is also consistent because moving to food decreases heuristic lower than actual cost. That keeps consistency of heuristic.

6)

In inadmissible heuristic finding solution can be faster than consistent (also admissible) one. Because we spend lots of time while finding heuristic cost at all steps. That’s why simple and inadmissible heuristic can find solution fast. However, there will be a high possibility that this result is not optimal. If time is great concern, we could choose inadmissible heuristic, Otherwise we should use consistent heuristic to find optimal path.