Ayça Avcı - 53961

Answer 1:

When using BFS in bigMaze, it expended 620 nodes and found path cost 210. When using BFS in openMaze, it expended 682 nodes and found path cost 456. When using DFS in bigMaze it expended 390 nodes and found path cost 210. When using DFS in openMaze it expended 576 nodes and found path cost 298. If memory consumption is a problem, we can use DFS over BFS. If we need an optimal solution, we should use BFS over DFS.

Answer 2:

Since I can be able to solve question 1, 2 and 8 only in the project by myself, I looked other questions' solution from GitHub.

According to code that I took, for openMaze, UCS expanded 682 nodes and cost of the path is 54, for bigMaze, it expanded 620 nodes and cost of path is 210. Again, for openMaze, A* expanded 535 nodes and cost of path is 54, for bigMaze, it expanded 549 and cost of path is 210. Difference between UCS and A* is heuristic. So, if we have a good heuristic, we should use A* over UCS, otherwise, UCS over A*.

Answer 3:

According to code that I took from GitHub, to check whether we visited the corners or not, storing only x,y coordinates are not enough. Additional to x,y coordinates, we need to hold four booleans to check whether we visited the corners or not.

Answer 4:

According to code that I took from GitHub, since costs are equal in current Pacman, Manhattan distance as a heuristic will always hold the admissibility condition, h(s) <= total cost to reach from s to g (goal state), considering we have no obstacle along the path which is best case scenario. Even there were obstacles along the path, condition will hold and total cost to reach from s to g will be always greater than h(s).

Answer 5:

According to code that I took from GitHub, we can also use Manhattan distance again as a heuristic just like the previous question since admissibility and consistency are held. In the previous question, we were looking at the distance between current state and corners, now we can also look at the distance between current state and foods with the same approach.

Answer 6:

Inadmissible heuristics do not give always optimal solutions (suboptimal solutions). Since consistency implies admissibility, consistent heuristics always give an optimal solution. If we have time or memory restrictions, in some cases, we can prefer to use inadmissible heuristics over consistent ones which at least give us a solution whereas we may not even get a solution using consistent heuristics. If we really need an optimal solution, we should use consistent heuristics over inadmissible ones.