Andrew Katsanevas
Bradley Dawn
1/17/2019

Project 1 Answers

3.1 Depth First Search

2. Is the exploration order what you would have expected? Does Pacman actually go to all the explored squares on his way to the goal?
The exploration order is what I expected for depth first search. There were many explored squares that Pacman did not actually go to. Most notably, there was an entire different path that was explored that Pacman didn't take.

3. Is this a least cost solution? If not, think about what depth-first search is doing wrong.
This is not the least cost solution. DFS does not necessarily find the shortest path because it expands the deepest nodes (using a stack). Pacman explored a longer path first, so he found the goal on that longer path before searching the shorter path.

3.2 Breadth First Search

2. Does BFS find a least cost solution? If so explain why?
Yes. BFS actually finds the shortest path because it expands the closest nodes (using a queue). This is why BFS is used to find shortest path rather than DFS.

3.3 Uniform Cost Search

2. Specify the data structure used from the util.py for the uniform cost search
For UCS, we used the PriorityQueueWithFunction from util.py. We passed it a lambda function that returns the getCostOfActions() cost of the given problem and path.

3.4 A* search

2. What happens on openMaze for the various search strategies? Describe your answer in terms of the solution you get for A* and Uniform cost search.
UCS searches every square, expanding 682 nodes. (cost 54, 0.1 sec, 682 nodes expanded),A*

ignores a couple large sections, expanding 535 nodes. Both of them find the same path score (456) in the end though, and have the same cost (54) and time (0.1 seconds).

3.5 Finding All the Corners

2. Describe in few words/lines the state representation you chose or how you solved the problem of finding all corners?
For our state representation, we chose to add the tuple of corners to the state in addition to the position. We then removed a corner from the tuple if we visited it. The goal is reached when the corners tuple is empty.

3.6 Corners Problem: Heuristic

2. Describe the heuristic you had used for the implementation.
We created a heuristic which returns the maximum Manhattan distance from any corner. It goes through the corners and finds the furthest one, returning it's distance from Pacman's position. This will always be lower than the shortest path to a goal.

3.7 Eating All Dots

2. Describe the heuristic you had used for the FoodSearchProblem.
We took the foodGrid as a list, and found the furthest food dot from Pacman in terms of mazeDistance. The heuristic is the maxDistance to the food from Pacman.

3.8 Suboptimal Search

2. Explain why the ClosestDotSearchAgent won't always find the shortest possible path through the maze.
Because it is a greedy approach to a problem that can't always be optimally solved by a greedy approach. As can be seen in bigSearch, Pacman may follow a path of dots that are one space away from him, ignoring some dot along the way. In the bigSearch example, he leaves a single dot behind on the right side of the maze while eating everything on the left side. Once the left side is finished, he travels all the way back to the right. In the shortest possible path, Pacman would pick up that dot while he was still on the right side.

4 Self Analysis

1. What was the hardest part of the assignment for you?
Trying various heuristic ideas until we found one that worked well. We tried at least a few heuristics that weren't admissible before figuring out good ones.

2. What was the easiest part of the assignment for you?
Implementing the search algorithms was fairly simple, since they all follow the same pattern except for their data structure. So all we needed to do was write a generic search that takes a data structure parameter.

3. What problem(s) helped further your understanding of the course material?
We think that it was a good lesson that all of these search algorithms follow the same procedure with different data structures. And the heuristic writing was good for learning what a consistent and admissible heuristic is.

4. Did you feel any problems were tedious and not helpful to your understanding of the Material?
We think every part of this assignment was pretty helpful to our understanding. Nothing was incredibly tedious, other than perhaps trying different heurisitics (but again that was helpful to our understanding).

5. What other feedback do you have about this homework?
It was fairly interesting, and representing it visually with Pacman makes things more easily understandable.