Maximillian Hennessey (mh1285)

02/01/22

CS730

1.

I thought I could use my old DFS data structure that performs cycle checking for the iterative deepening searching. I quickly found that I needed to make adjustments to this. The first iteration to the cyclce checking (what I used in DFS) was not removing any of the nodes once I started moving back up the tree in my recursion causing the search to end after only exploring one branches depth because the root node was always included. The second iteration I used the same data structure but tried to clear it every time I would go back up the tree, this led to essentially no cycle checking and causing infinite loops. Finally, I changed to a hashtable of a hash table and. Working the same as the closed list hashtable I implemented with the other search algorithm, but the nodes are hashed by depth. This led me to faster times and admissible results, but was now basically BFS. My first implementation actually worked, I just needed to clear/reset the underlying data structure every iteration with a different max-depth.

2.

My first heuristic, simply the distance an action takes, in this case 1, this makes this similar to BFS which is admissible. My second heuristic, h2, is the Manhattan distance of the agent to the closest piece of dirt. In all tests for h2, my solution was admissible, because it's impossible to take less moves to get to any location quicker than directly moving to the closest one. However, this can fail when the number of walls is very large, due to the distance not accounting for obstacles, so the only way to fix this is by giving walls a large cost when calculating the Manhattan distance.

3.

Iterative deepening time complexity = O(b^d) d = solution depth (same as BFS) **admissible**

Iterative deepening space complexity = O(bd) d = solution depth (same as DFS) **admissible**

A* time complexity (h = 0 or 1) = O(b^d) d = solution depth (same as BFS) **admissible**

A* space complexity (h = 0 or 1) = O(b^d) d = solution depth (same as BFS) **admissible**

A* time complexity (h = n) = O(n) admissible

A* space complexity (h = n) = O(n) admissible

4.

For iterative deepening: IDS will continue to search till it gets to a depth with the solution, therefore it will exponentially expand nodes same as BFS (O(b^d)), but because as the algorithm traverses back up the tree it reduces its size O(bd).

For A* h=0 or 1: When using a constant heuristic, A* will become a BFS (uniform cost) as the nodes will still expand in an order as when expanding nodes their new cost (cost = g + h) will be uniform. This will give it the same complexities as BFS.

For A* h=Manhattan distance: The Manhattan distance is an admissible heuristic because the agent will have to be move to least the total cost every iteration.

5.

The reference provided (vacuum-plan-reference) failed to run IDS on somewhat large worlds, if that is intentional, I feel it should be mentioned in the assignment.

Also, I felt that the directions for finding a heuristic were fairly vague. I would like to maybe have some more direction on how we should select our heuristic.