Homework #1

Oregon State University

CS331 Artificial Intelligence

Spring 2016

Student:
Sam Quinn
Quinnsa@Oregonstate.edu

Professor:
Dr. Rebecca Hutchinson
Rah@eecs.oregonstate.edu

April 15, 2016

Oregon State
UNIVERSITY

College of Engineering

Contents

1 Methodology

Usage: python quinnsaHW1.py –help
Breadth First Search: python quinnsaHW1.py –mode bfs –init start1.txt –goal goal1.txt
Depth First Search: python quinnsaHW1.py –mode dfs –init start1.txt –goal goal1.txt
Iterative Deepening Depth First Search: python quinnsaHW1.py –mode iddfs –init start1.txt –goal goal1.txt
A-Star Search: python quinnsaHW1.py –mode astar –init start1.txt –goal goal1.txt

For this assignment I tested my program on the three scenarios provided with each of my search algorithms.

1.1 BFS

The BFS search function will explore every node in the tree level by level. This algorithm will insert all of the neighbor nodes to the end of the data structure ensuring that it will explore each node and its neighbors before continuing to the next level of the tree.

1.2 DFS

The DFS search function is implemented very similarly to the BFS search function but instead of adding the neighbor nodes to the end of the data structure the neighbors are added to the front. With each node's neighbors added to the front the algorithm will continue to explore one path until a terminal.

1.3 IDDFS

The Iterative Deepening Depth First Search will start off with the root node and based on the depth limit perform a depth first search. If no solution is found the limit will increase and the whole process will start over again. This algorithm mimics a breadth first search in some sense.

1.4 A-Star

The A-Star search algorithm is the only informed search algorithm that I implemented. The A-Star search requires a heuristic and a priority queue. The heuristic has to be admissible, essentially meaning that it is "best case". For my heuristic I summed the number of Humans, Cannibals, and the boat then subtracted the number of Humans, Cannibals, and boat from the left side resulting in the heuristic value. The heuristic value that is returned will be where that number is placed within the data structure, for example when a heuristic value is 0 (goal node) it will be

placed at the index of 0. The A-Star function will pop nodes from the front of the data structure, meaning that the best nodes from the heuristic function will be analyzed first.

2 Results

	BFS	**DFS**	**IDDFS**	**A-Star**
Test 1	Solution: 12 Expanded: 16	Solution: 12 Expanded: 16	Solution: 14 Expanded: 102	Solution: 15 Expanded: 28
Test 2	Solution: 34 Expanded: 57	Solution: 34 Expanded: 57	Solution: 35 Expanded: 514	Solution: 42 Expanded: 82
Test 3	Solution: 1514 Expanded: 2254	Solution: 1514 Expanded: 2254	Solution: 494 Expanded: 384086	Solution: 1611 Expanded: 6438

3 Discussion

After running my program on all three test cases I was surprised to see the results. It was interesting to see that DFS and BFS had the same solution path number as well as the same number of nodes expanded. BFS and DFS out performed IDDFS and A-Star in Test 1 and 2 but IDDFS really pulled ahead on Test 3 getting the solution in less than half of the other algorithms. A-Star was the worst in all the tests, I think this is to do with my heuristic function not optimizing the best as possible. The A-Star heuristic, since really didn't do too much thinking and was admissible might have prioritized valuable nodes incorrectly.

IDDFS took a very long time on the largest data set (test 3), almost two full minute to find the solution even though it found the best solution it expanded 384086 nodes which is much much more than any other algorithm.

4 Conclusion

This was very interesting testing the performance of these algorithms, this was my first real hands on with search algorithms. What I can conclude from these results is that there really isn't one search algorithm that can solve everything. If you are looking for an algorithm that is fast and is guaranteed to find a solution you should use BFS. If you need an algorithm that is memory conscious you should use DFS. And if you need the most optimized path use IDDFS or A-Star with a good heuristic. It is hard to determine which algorithm performed the best because they all have unique pros and cons associated with each. But for me given this problem I think that I would choose IDDFS since it will give the most optimized path even though it takes much longer. I was not expecting to think that IDDFS would be the best because I thought that BFS would have similar results.