Written Documentation Assignment 3

Adam Terhaerdt
CS-380

This program was based off of Assignment 2. New commands that were added for this assignment were bfs, dfs, ids, and astar. When running these commands, they will run different search algorithms depending on which command is selected. They all use the same underlying function `search` which takes a board, a data structure, and potentially a depth limit (for IDS). The data structure of this function is dynamic based on which search algorithm you choose. DFS and IDS both use a stack, BFS uses a queue, and astar uses a priority queue (found in the main function). The main reason I wanted these search algorithms to be one function was to help test functionality. I made my own data structure files for all the mentioned data structures to ensure consistency of calling various functions for my data structures. In the search function, an optional depth_limit can be passed to search. If set to 1, the depth will be tracked to perform IDS. The check `if depth_limit is not None` is used to check if IDS is being used as this is the only algorithm that relies on the current depth. If the data structure is not empty, my program will determine if it needs to get priority, for A*, based on the data it pops. I also check to see if the data_structure is a stack and then I will reverse the available moves to ensure that my program picks the rightmost node to the left most node.

For analyzing the time and space complexity of the different searches I used the input filr “SBP-level2.txt”. BFS was able to find a solution length of 33 moves, exploring 723 nodes in 0.11 seconds while DFS was able to find a solution length of 190 moves, exploring 229 nodes in 0.02 seconds. You can see that DFS was quicker and explored less nodes than BFS, being much more memory efficient, but the algorithm found a much less optimal solution with a difference of 157 nodes from BFS. IDS was able to find a solution length of 93 nodes, exploring 5915 nodes, in 0.11 seconds. IDS was able to find a solution in between BFS and DFS, but explored a significant amount more nodes than both and took significantly longer to complete, taking 5.23 seconds. Finally A* was able to find an optimal solution of 33 moves, with only exploring 700 nodes, 23 nodes less than BFS, and was able to find a solution in 0.16 seconds.

These results are expected since BFS guarantees the shortest path in unweighted graphs and A* is optimal when using an admissible heuristic. DFS being the fastest makes sense as it uses the least memory to find an not-optimal solution. IDS explored significantly more nodes and took the longest time to find a suboptimal solution. The heuristic that I used for A* was the Manhattan distance.

Results:

./run.sh bfs "../SBP-levels/SBP-level0.txt"

(2,LEFT)
(4,DOWN)
(3,RIGHT)
(2,UP)
(2,UP)

5, 4
 1, 2, 2, 1, 1
 1, 0, 0, 3, 1
 1, 0, 0, 4, 1
 1, 1, 1, 1, 1

15
0.00
5

./run.sh bfs "../SBP-levels/SBP-level2.txt"

(3,LEFT)
(4,LEFT)
(7,UP)
(7,RIGHT)
(2,LEFT)
(6,UP)
(8,UP)
(8,RIGHT)
(5,DOWN)
(7,LEFT)
(5,LEFT)
(5,DOWN)
(2,DOWN)
(4,LEFT)
(5,UP)
(2,RIGHT)
(3,RIGHT)
(6,UP)
(7,LEFT)
(8,LEFT)

(6,UP)
(7,UP)
(8,LEFT)
(2,DOWN)
(6,DOWN)
(5,RIGHT)
(4,RIGHT)
(6,RIGHT)
(6,UP)
(8,UP)
(2,LEFT)
(2,LEFT)
(2,LEFT)

6, 5
1, 1, 1, 1, 1, 1
1, 3, 4, 5, 6, 1
1, 7, 7, 5, 8, 1
2, 2, 0, 0, 0, 1
1, 1, 1, 1, 1, 1

723
0.15
33

./run.sh dfs "../SBP-levels/SBP-level0.txt"
(2,LEFT)
(3,LEFT)
(2,RIGHT)
(3,DOWN)
(3,LEFT)
(3,LEFT)
(2,UP)
(4,RIGHT)
(3,DOWN)
(2,LEFT)
(2,UP)

5, 4

1, 2, 2, 1, 1
1, 0, 0, 0, 1
1, 3, 4, 0, 1
1, 1, 1, 1, 1

11
0.00
11

./run.sh dfs "../SBP-levels/SBP-level2.txt"
(3,LEFT)
(2,LEFT)
(4,LEFT)
(2,RIGHT)
(4,UP)
(5,LEFT)
(4,DOWN)
(2,LEFT)
(8,UP)
(2,RIGHT)
(4,UP)
(5,LEFT)
(6,LEFT)
(8,RIGHT)
(6,RIGHT)
(5,RIGHT)
(4,DOWN)
(2,LEFT)
(6,UP)
(6,RIGHT)
(5,RIGHT)
(5,DOWN)
(2,DOWN)
(4,LEFT)
(4,LEFT)
(5,UP)
(2,RIGHT)
(4,RIGHT)
(2,LEFT)
(8,UP)

(4,LEFT)
(5,LEFT)
(6,UP)
(2,RIGHT)
(4,DOWN)
(4,LEFT)
(5,LEFT)
(8,RIGHT)
(5,RIGHT)
(4,RIGHT)
(6,UP)
(2,LEFT)
(6,DOWN)
(5,RIGHT)
(4,RIGHT)
(7,RIGHT)
(3,DOWN)
(3,LEFT)
(3,LEFT)
(2,UP)
(6,LEFT)
(4,DOWN)
(2,RIGHT)
(3,RIGHT)
(3,DOWN)
(2,LEFT)
(2,LEFT)
(5,UP)
(3,RIGHT)
(2,RIGHT)
(4,UP)
(5,RIGHT)
(3,DOWN)
(2,LEFT)
(3,LEFT)
(6,UP)
(6,RIGHT)
(7,UP)
(8,LEFT)
(7,DOWN)

(4,DOWN)
(3,RIGHT)
(2,RIGHT)
(4,UP)
(7,LEFT)
(3,DOWN)
(2,LEFT)
(3,LEFT)
(6,UP)
(6,RIGHT)
(7,UP)
(8,LEFT)
(6,DOWN)
(6,LEFT)
(4,DOWN)
(3,RIGHT)
(2,RIGHT)
(4,UP)
(6,DOWN)
(3,DOWN)
(2,LEFT)
(3,LEFT)
(5,RIGHT)
(3,RIGHT)
(2,RIGHT)
(4,UP)
(6,LEFT)
(5,LEFT)
(4,DOWN)
(2,RIGHT)
(7,LEFT)
(2,LEFT)
(5,UP)
(5,RIGHT)
(7,UP)
(7,RIGHT)
(3,DOWN)
(2,LEFT)
(3,LEFT)
(8,LEFT)

(3,RIGHT)
(2,RIGHT)
(4,UP)
(7,LEFT)
(5,DOWN)
(5,LEFT)
(4,DOWN)
(2,RIGHT)
(5,DOWN)
(2,LEFT)
(4,RIGHT)
(2,RIGHT)
(3,RIGHT)
(5,UP)
(4,UP)
(6,LEFT)
(3,DOWN)
(3,DOWN)
(2,LEFT)
(2,LEFT)
(5,UP)
(7,UP)
(6,RIGHT)
(5,RIGHT)
(4,RIGHT)
(4,DOWN)
(2,DOWN)
(3,LEFT)
(3,LEFT)
(4,UP)
(5,UP)
(4,DOWN)
(3,RIGHT)
(8,UP)
(3,LEFT)
(4,LEFT)
(6,UP)
(6,RIGHT)
(2,RIGHT)
(7,UP)

(8,RIGHT)
(6,DOWN)
(2,LEFT)
(4,DOWN)
(3,RIGHT)
(7,RIGHT)
(3,LEFT)
(4,LEFT)
(6,UP)
(7,LEFT)
(5,DOWN)
(8,LEFT)
(4,RIGHT)
(3,RIGHT)
(5,DOWN)
(2,RIGHT)
(3,LEFT)
(2,LEFT)
(4,LEFT)
(2,RIGHT)
(5,UP)
(2,LEFT)
(4,DOWN)
(3,RIGHT)
(8,RIGHT)
(3,LEFT)
(5,DOWN)
(2,RIGHT)
(3,RIGHT)
(5,UP)
(3,LEFT)
(6,LEFT)
(3,RIGHT)
(5,UP)
(2,LEFT)
(6,RIGHT)
(7,UP)
(7,RIGHT)
(2,DOWN)
(2,LEFT)

6, 5
 1, 1, 1, 1, 1, 1
 1, 3, 4, 4, 5, 1
 1, 0, 0, 6, 5, 1
 2, 2, 0, 7, 8, 1
 1, 1, 1, 1, 1, 1

229
0.03
190

./run.sh ids "../SBP-levels/SBP-level0.txt"
(2,LEFT)
(3,LEFT)
(4,DOWN)
(3,RIGHT)
(3,RIGHT)
(2,UP)
(2,UP)

5, 4
 1, 2, 2, 1, 1
 1, 0, 0, 3, 1
 1, 0, 0, 4, 1
 1, 1, 1, 1, 1

20
0.01
7

./run.sh ids "../SBP-levels/SBP-level2.txt"
(3,LEFT)
(2,LEFT)
(4,LEFT)
(2,RIGHT)
(4,UP)
(5,LEFT)

(4,DOWN)
(2,LEFT)
(8,UP)
(2,RIGHT)
(4,UP)
(5,LEFT)
(6,LEFT)
(8,RIGHT)
(6,RIGHT)
(5,RIGHT)
(4,DOWN)
(2,LEFT)
(6,UP)
(6,RIGHT)
(5,RIGHT)
(5,DOWN)
(2,DOWN)
(4,LEFT)
(4,LEFT)
(5,UP)
(2,RIGHT)
(4,RIGHT)
(2,LEFT)
(8,UP)
(4,LEFT)
(5,LEFT)
(6,UP)
(2,RIGHT)
(4,DOWN)
(4,LEFT)
(5,LEFT)
(8,RIGHT)
(5,RIGHT)
(4,RIGHT)
(6,UP)
(2,LEFT)
(6,DOWN)
(5,RIGHT)
(4,RIGHT)
(7,RIGHT)

(3,DOWN)
(3,LEFT)
(3,LEFT)
(4,LEFT)
(3,RIGHT)
(5,UP)
(6,UP)
(8,UP)
(3,DOWN)
(3,LEFT)
(4,LEFT)
(5,LEFT)
(7,UP)
(2,RIGHT)
(8,RIGHT)
(7,RIGHT)
(3,DOWN)
(3,LEFT)
(4,LEFT)
(5,LEFT)
(6,DOWN)
(3,DOWN)
(3,LEFT)
(4,RIGHT)
(3,RIGHT)
(3,RIGHT)
(5,UP)
(6,UP)
(3,DOWN)
(8,LEFT)
(6,UP)
(7,UP)
(8,LEFT)
(2,DOWN)
(5,DOWN)
(5,LEFT)
(7,RIGHT)
(5,RIGHT)
(4,RIGHT)
(3,RIGHT)

(3,DOWN)
(5,UP)
(6,UP)
(8,UP)
(2,LEFT)
(2,LEFT)
(2,LEFT)

6, 5
 1, 1, 1, 1, 1, 1
 1, 3, 4, 5, 6, 1
 1, 7, 7, 5, 8, 1
 2, 2, 0, 0, 0, 1
 1, 1, 1, 1, 1, 1

5915
5.42
93

./run.sh astar "../SBP-levels/SBP-level0.txt"
(2,LEFT)
(4,DOWN)
(3,RIGHT)
(2,UP)
(2,UP)

5, 4
 1, 2, 2, 1, 1
 1, 0, 0, 3, 1
 1, 0, 0, 4, 1
 1, 1, 1, 1, 1

12
0.00
5

./run.sh astar "../SBP-levels/SBP-level2.txt"

(3,LEFT)
(2,LEFT)
(5,UP)
(8,UP)
(5,LEFT)
(8,UP)
(8,RIGHT)
(8,RIGHT)
(5,DOWN)
(7,LEFT)
(5,LEFT)
(5,DOWN)
(2,DOWN)
(4,LEFT)
(5,UP)
(2,RIGHT)
(3,RIGHT)
(6,UP)
(7,LEFT)
(8,LEFT)
(6,UP)
(7,UP)
(8,LEFT)
(2,DOWN)
(6,DOWN)
(5,RIGHT)
(4,RIGHT)
(6,RIGHT)
(6,UP)
(8,UP)
(2,LEFT)
(2,LEFT)
(2,LEFT)

6, 5
1, 1, 1, 1, 1, 1
1, 3, 4, 5, 6, 1
1, 7, 7, 5, 8, 1
2, 2, 0, 0, 0, 1

1, 1, 1, 1, 1, 1

700
0.00
33