CMPE480 - Project1 Report

> Ahmet Gedemenli

> 2014400147

Solution

I simply created a unique state ID for each state. State IDs are 5 digit numbers in my solution. The first two digits are the row id of our location, the second two digits are the column id of our location, and the last digit is the **orientation**. It means we are not only on this cell, but also on the right next one if the last digit is 1; the upper next cell if the last digit is 2; and no orientation(single cell) if the last digit is 0.

After I created the state IDs I implemented functions for each different search algorithm. I also used many helper functions, which are shortly explained with comments in the code.

Heuristics

My heuristics is the manhattan distance between the current location and the goal state's location. I am also adding the orientation difference to it. Because of that, `h() = 0` when are on the goal state, but `h() = 1` when we are on the goal state's location with an orientation. It is not a perfect heuristic. Even though it makes a significant impact on cases like level-1; it may not be so useful in some tricky cases like level-4, which I will discuss below. However, since it is a heuristic approach, we can't expect it to be perfectly accurate, but it helps us in many cases.

Evaluation and Comparison

As can be seen in the sample results below, the least efficient search algorithm is DFS in both of our cases. UCS and A* gives the optimal solution because they are proven to be optimal. BFS finds a sub-optimal solution which is really close to the optimal one. I believe this is because the costs are not so different, they are all 1 and 3. If the cost difference were larger, then BFS would be far away from the optimal solution. Greedy search finds a not so bad solution, thanks to our heuristic function.

I really enjoyed coding this project. It was a good hands on experience for me. I found A* to be the best search algorithm amongst all, since it finds the optimal solution with very small explores thanks to our heuristics function. `f()=g()+h'()` gives us a really good estimation since it is a combination of a real measurement with a heuristic.

Level 1 Outputs

./proj1 levels/level1.txt dfs

105 72 63 63

RRDLLULURDLLURDRRDLLLURRDRURDRRRDLULULLDRRRULDLURRDLULLLDRRRRDL

./proj1 levels/level1.txt bfs

11 35 7 7

RRDRRRD

./proj1 levels/level1.txt ucs

10 22 8 8

DRRRRRRD

./proj1 levels/level1.txt gs

17 33 21 11

DRRRRDRRULD

./proj1 levels/level1.txt as

10 8 8 8

DRRRRRRD

Level 4 Outputs

./proj1 levels/level4.txt dfs

102 60 60 60

ULDRULDRURULDLURRDLULDRRRRUURRRDRULLLLDRURRRDDRDLLUURDRDLDRU

./proj1 levels/level4.txt bfs

33 83 19 19

ULDRURRRRUURRRDDDRU

./proj1 levels/level4.txt ucs

33 80 19 19

ULDRURRRRUURRRDDDRU

./proj1 levels/level4.txt gs

49 62 29 29

RURRRULDRUURRRDDRDLURRDLURDLU

./proj1 levels/level4.txt as

33 79 19 19

ULDRURRRRUURRRDDDRU