

COMS 573 Lab 1
Eranda Sooriyarachchi
Sep 30 2022

For this project I used the code provided by the textbook. I used Python for coding. The file **search.py** was edited to function properly for the lab1 requirements. The file **cs572lab1.py** was created and used for Part 1. The file **lab1part3.py** was created and used for both Part 2 and Part3 of the lab.

Breadth first tree search, IDS and A* searches with 3 heuristics were used. H1 was A* linear, H2 was A* Manhattan and H3 was A* Max_heuristic. In the printed report rather than saying H1,H2 or H3, I have included A* linear, A* Manhattan and A* Max_heuristic. The Max_Heuristic takes the best of H1 and H2.

Please Note that, if a particular algorithm has not printed any results on a particular file, that means the algorithm has surpassed the 15 minute timeout.

The format of all output files would be as follows:

File path

Nodes generated by Algorithm1

Time taken in ms and the name of the algorithm1

Path length

The steps of the path as UP, DOWN, LEFT, RIGHT

Nodes generated by Algorithm2

Time taken in ms and the name of the algorithm2

Path length

The steps of the path as UP, DOWN, LEFT, RIGHT

Nodes generated by Algorithm3

Time taken in ms and the name of the algorithm3

Path length

The steps of the path as UP, DOWN, LEFT, RIGHT

Nodes generated by Algorithm3

Time taken in ms and the name of the algorithm3

Path length

The steps of the path as UP, DOWN, LEFT, RIGHT

Please note the file path is mentioned at the top only once and then after that results of each algorithm for that particular file is printed.

In the part 1 output, since only one algorithm is run on only one file, it will be the file name and the results directly below.

All results of Part2 and Part3 are in .txt files in a Folder named results.

Part 1

To run a particular algorithm on a txt file with the puzzle, enter the following in the terminal:

```
python3 cs572lab1.py ./Part2/S4.txt BFS
```

This will give the following output:

```
./Part2/S4.txt
```

```
Generated nodes: 181347
```

```
Time taken for BFS = 0.47585556292533876 ms
```

```
Path length: 31
```

```
['UP', 'UP', 'LEFT', 'DOWN', 'DOWN', 'RIGHT', 'RIGHT', 'UP', 'UP', 'LEFT', 'DOWN', 'LEFT', 'DOWN',  
'RIGHT', 'RIGHT', 'UP', 'UP', 'LEFT', 'DOWN', 'LEFT', 'DOWN', 'RIGHT', 'RIGHT', 'UP', 'UP', 'LEFT',  
'LEFT', 'DOWN', 'DOWN', 'RIGHT', 'RIGHT']
```

Part 2

To run all the algorithms on all puzzles in Part 2 you should enter the following in the terminal:

```
python3 lab1part3.py ./Part2 > Part2.txt
```

Results

```
/Users/erandasooriyarachchi/Downloads/Part2/S4.txt
```

```
Generated nodes: 181347
```

```
Time taken for BFS = 475.85556292533876 s
```

```
Path length: 31
```

```
['UP', 'UP', 'LEFT', 'DOWN', 'DOWN', 'RIGHT', 'RIGHT', 'UP', 'UP', 'LEFT', 'DOWN', 'LEFT', 'DOWN',  
'RIGHT', 'RIGHT', 'UP', 'UP', 'LEFT', 'DOWN', 'LEFT', 'DOWN', 'RIGHT', 'RIGHT', 'UP', 'UP', 'LEFT',  
'LEFT', 'DOWN', 'DOWN', 'RIGHT', 'RIGHT']
```

```
Number of Generated Nodes: 17489
```

```
Time taken for A* manhattan = 43.396644115447996 s
```

```
Path length: 31
```

```
['RIGHT', 'UP', 'UP', 'LEFT', 'LEFT', 'DOWN', 'DOWN', 'RIGHT', 'RIGHT', 'UP', 'LEFT', 'LEFT', 'DOWN',  
'RIGHT', 'RIGHT', 'UP', 'UP', 'LEFT', 'DOWN', 'RIGHT', 'UP', 'LEFT', 'LEFT', 'DOWN', 'DOWN',  
'RIGHT', 'RIGHT', 'UP', 'LEFT', 'DOWN', 'RIGHT']
```

Number of Generated Nodes: 17489

Time taken for A* max_heuristics = 42.16121792793274 s

Path length: 31

['RIGHT', 'UP', 'UP', 'LEFT', 'LEFT', 'DOWN', 'DOWN', 'RIGHT', 'RIGHT', 'UP', 'LEFT', 'LEFT', 'DOWN', 'RIGHT', 'RIGHT', 'UP', 'UP', 'LEFT', 'DOWN', 'RIGHT', 'UP', 'LEFT', 'LEFT', 'DOWN', 'DOWN', 'RIGHT', 'RIGHT', 'UP', 'LEFT', 'DOWN', 'RIGHT']

/Users/erandasooriyarachchi/Downloads/Part2/S5.txt

Generated nodes: 54

Time taken for BFS = 6.01053237915039e-04 s

Path length: 6

['LEFT', 'UP', 'RIGHT', 'DOWN', 'DOWN', 'RIGHT']

Number of Generated Nodes: 88

Time taken for IDS = 2.3250579833984377e-03 s

Path length: 6

['LEFT', 'UP', 'RIGHT', 'DOWN', 'DOWN', 'RIGHT']

Number of Generated Nodes: 9

Time taken for A*linear = 1.4209747314453124e-04 s

Path length: 6

['LEFT', 'UP', 'RIGHT', 'DOWN', 'DOWN', 'RIGHT']

Number of Generated Nodes: 10

Time taken for A* manhattan = 2.288818359375e-04 s

Path length: 6

['LEFT', 'UP', 'RIGHT', 'DOWN', 'DOWN', 'RIGHT']

Number of Generated Nodes: 10

Time taken for A* max_heuristics = 2.5010108947753905e-04 s

Path length: 6

['LEFT', 'UP', 'RIGHT', 'DOWN', 'DOWN', 'RIGHT']

/Users/erandasooriyarachchi/Downloads/Part2/S1.txt

Generated nodes: 97527

Time taken for BFS = 285.4613687992096 s

Path length: 24

['UP', 'UP', 'RIGHT', 'DOWN', 'DOWN', 'RIGHT', 'UP', 'LEFT', 'LEFT', 'DOWN', 'RIGHT', 'RIGHT', 'UP', 'LEFT', 'LEFT', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'LEFT', 'DOWN', 'RIGHT', 'RIGHT']

Number of Generated Nodes: 12498

Time taken for A*linear = 20.97235083580017 s

Path length: 24

['UP', 'UP', 'RIGHT', 'DOWN', 'DOWN', 'RIGHT', 'UP', 'LEFT', 'LEFT', 'DOWN', 'RIGHT', 'RIGHT', 'UP', 'LEFT', 'LEFT', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'LEFT', 'DOWN', 'RIGHT', 'RIGHT']

Number of Generated Nodes: 2025

Time taken for A* manhattan = 0.6414971351623535 s

Path length: 24

['UP', 'UP', 'RIGHT', 'DOWN', 'DOWN', 'RIGHT', 'UP', 'LEFT', 'LEFT', 'DOWN', 'RIGHT', 'RIGHT', 'UP', 'LEFT', 'LEFT', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'LEFT', 'DOWN', 'RIGHT', 'RIGHT']

Number of Generated Nodes: 2025

Time taken for A* max_heuristics = 0.6446480751037597 s

Path length: 24

['UP', 'UP', 'RIGHT', 'DOWN', 'DOWN', 'RIGHT', 'UP', 'LEFT', 'LEFT', 'DOWN', 'RIGHT', 'RIGHT', 'UP', 'LEFT', 'LEFT', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'LEFT', 'DOWN', 'RIGHT', 'RIGHT']

/Users/erandasooriyarachchi/Downloads/Part2/S2.txt

Generated nodes: 29053

Time taken for BFS = 38.67422413825989 s

Path length: 20

['UP', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'DOWN', 'RIGHT', 'UP', 'LEFT', 'LEFT', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'LEFT', 'DOWN', 'RIGHT', 'RIGHT']

Number of Generated Nodes: 1855

Time taken for A*linear = 0.5116300582885742 s

Path length: 20

['UP', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'DOWN', 'RIGHT', 'UP', 'LEFT', 'LEFT', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'LEFT', 'DOWN', 'RIGHT', 'RIGHT']

Number of Generated Nodes: 196

Time taken for A* manhattan = 9.715080261230468e-03 s

Path length: 20

['UP', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'DOWN', 'RIGHT', 'UP', 'LEFT', 'LEFT', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'LEFT', 'DOWN', 'RIGHT', 'RIGHT']

Number of Generated Nodes: 196

Time taken for A* max_heuristics = 1.0117053985595703e-02 s

Path length: 20

['UP', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'DOWN', 'RIGHT', 'UP', 'LEFT', 'LEFT', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'LEFT', 'DOWN', 'RIGHT', 'RIGHT']

/Users/erandasooriyarachchi/Downloads/Part2/S3.txt

puzzle is unsolvable

BFS Average time: 88.8879729906718

IDS Average time: 2.5833977593315973e-04

A* H1 Average time: 2.38712477684021

A* H2 Average time: 4.8942316903008345

A* H3 Average time: 4.757359239790175

Part 3

To run all algorithm on all text files in a folder with the puzzle

To run all the algorithms on all files in L8:

python3 lab1part3.py ./Part3/L8

To run all the algorithms on all files in L15:

python3 lab1part3.py ./Part3/L15

To run all the algorithms on all files in L24:

```
python3 lab1part3.py ./Part3/L24
```

If you want to save results into a .txt file:

```
python3 lab1part3.py ./Part3/L8 > L8.txt
```

```
python3 lab1part3.py ./Part3/L15 > L15.txt
```

```
python3 lab1part3.py ./Part3/L24 > L24.txt
```

Part3

Depth	BFS		IDS		A* : h1		A* : h2		A* : h3	
	Avg.run time	Avg #nodes Explored	Avg.run time	Avg #nodes Explored	Avg.run time	Avg #nodes Explored	Avg.run time	Avg #nodes Explored	Avg.run time	Avg #nodes Explored
8	1.13E-03	134.6	5.25E-03	209.65	1.06E-04	13.9	1.37E-04	12.5	1.54E-04	12.5
15	0.57692919	85302	8.55490434	5869.3	8.17E-03	282.95	2.42E-03	116	2.56E-03	115.85
24	154.907248	107117.7	Time out		11.0939443	12899.2	0.1847671	1483.6	0.00018645	1483.6

From the results of part 3 that have been tabulated, we can observe that, as the depth increases the time taken per run on average increase in all search algorithms. The time taken by IDS increases the most on average, which is why when the depth was 24, the algorithm timed out for all files. A*: h2, which is the A* with Manhattan heuristic is the fastest algorithm. Then the A* H3 which again uses the fastest of the two A* searches is the second fastest, Even A* Linear which is A* h1, is considerably fast. BFS-graph search is better than IDS but at least 10-fold slower than the slowest A* search algorithm tested.

So overall, IDS is the slowest algorithm. A*h2 which is the A* with Manhattan heuristic is the fastest search algorithm.