Lab3

Name: Sushant Gautam (sg2223)

For **Part A and Part B** solution, see the astar.py.

Below is the output generated of partB implementation.

Part C:

Modify the priority calculation to have the search run Greedy Best-first search and Uniform cost search for all of the 5 maps. For each type of search, report the solution quality, the time it took for the search to run, and the number of nodes expanded in the search. Also show the path that was generated for each of these searches. In your report, discuss what are the advantages and disadvantages of each search algorithm on these maps and discuss which algorithm performs the best.

→ I modified the code to include priority calculation to have the search run Greedy Best-first search and Uniform cost search for all the 5 maps. Below is the comparison output of each algorithm.

Run Command: **python astar.py world/world.world ASTAR** for A* star search,

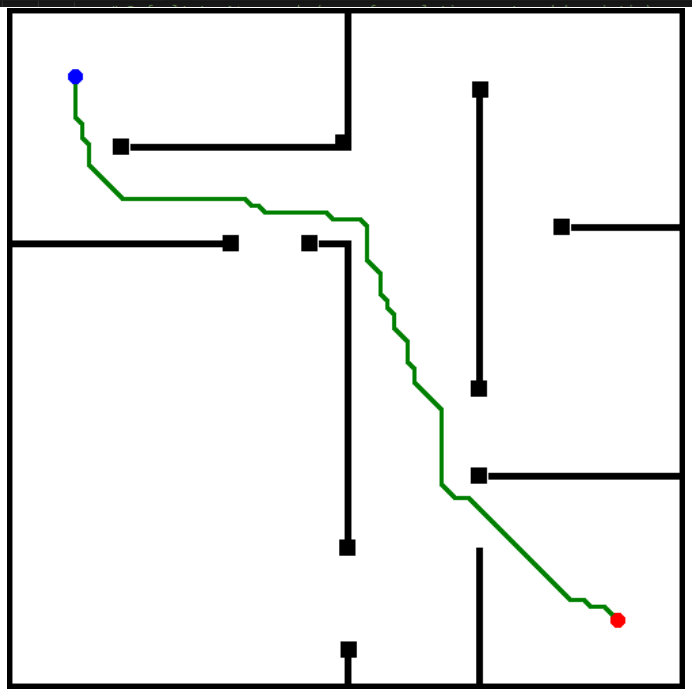
python astar.py world/world.world GBFS for Greedy Best-first search and

python astar.py world/world.world UCS for Uniform cost search.

1) **A* Search**

1.1. For house.world : **python astar.py world/house.world ASTAR**

```
Exploring neighbor (9, 14)
Exploring neighbor (9, 16)
Exploring neighbor (9, 14)
Exploring neighbor (10, 13)
Exploring neighbor (9, 13)
Exploring neighbor (9, 15)
Exploring neighbor (9, 13)
Exploring neighbor (10, 12)
Exploring neighbor (9, 12)
Exploring neighbor (9, 14)
Exploring neighbor (9, 12)
Exploring neighbor (10, 11)
Exploring neighbor (9, 11)
Exploring neighbor (9, 13)
Exploring neighbor (11, 11)
Exploring neighbor (9, 11)
Exploring neighbor (11, 11)
Exploring neighbor (10, 10)
Exploring neighbor (9, 10)
Exploring neighbor (9, 12)
Exploring neighbor (11, 10)
A* algorithm: 2220 node expansions
Quality/Cost of solution:  20.28928087696081
Search took:  0.4439370632171631 seconds
Saved image of path to file:  world/house_astarpath_ASTAR.png
(base) sushant@sg:~/catkin_ws/src/ai_labs$
```



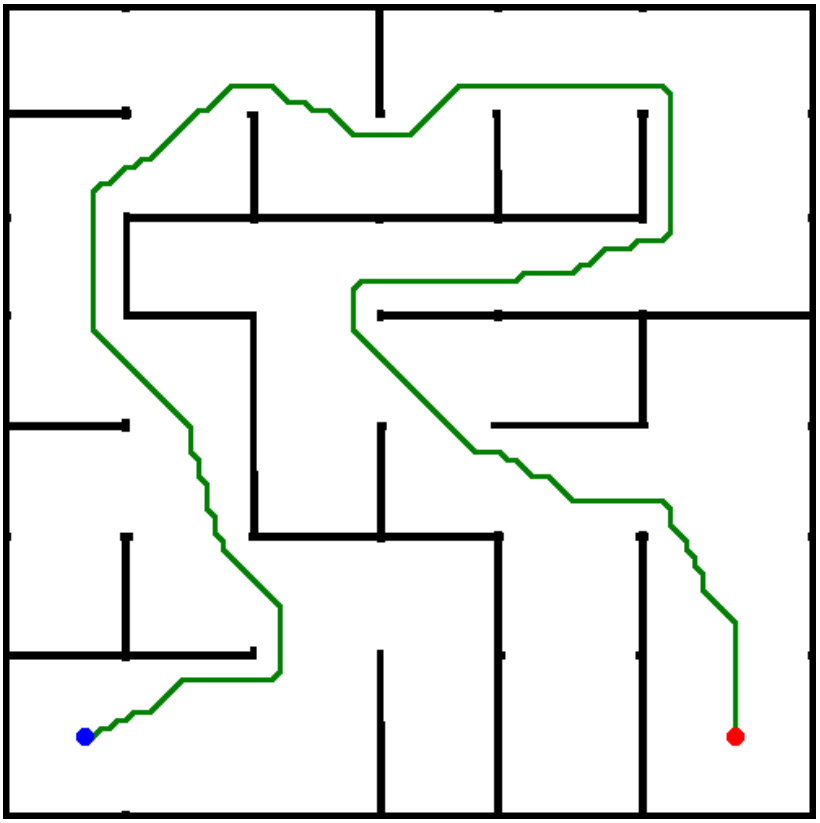
A* algorithm: 2220 node expansions

Quality/Cost of solution: 20.28928087696081

Search took: 0.4439370632171631 seconds

1.2. For maze1.world : **python astar.py world/maze1.world ASTAR**

```
Exploring neighbor (14, 86)
Exploring neighbor (13, 86)
Exploring neighbor (14, 85)
Exploring neighbor (13, 85)
Exploring neighbor (15, 85)
Exploring neighbor (14, 85)
Exploring neighbor (15, 84)
Exploring neighbor (14, 84)
Exploring neighbor (16, 84)
Exploring neighbor (15, 84)
Exploring neighbor (16, 83)
Exploring neighbor (15, 83)
Exploring neighbor (17, 83)
Exploring neighbor (16, 83)
Exploring neighbor (10, 90)
Exploring neighbor (12, 90)
Exploring neighbor (11, 89)
Exploring neighbor (11, 91)
Exploring neighbor (10, 89)
Exploring neighbor (10, 91)
Exploring neighbor (12, 91)
A* algorithm: 5259 node expansions
Quality/Cost of solution:  48.097167266400206
Search took:  2.2389657497406006 seconds
Saved image of path to file:  world/maze1_astarpath_ASTAR.png
```



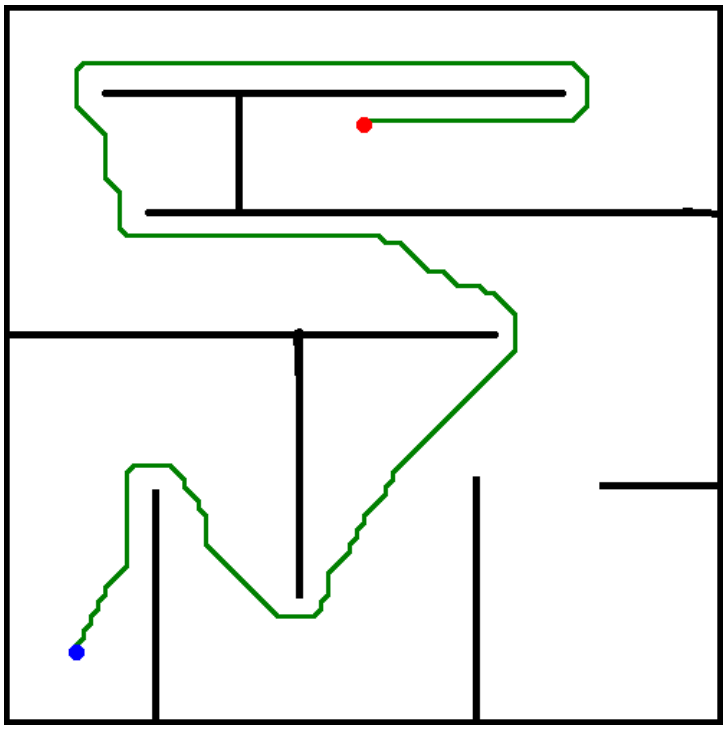
A* algorithm: 5259 node expansions

Quality/Cost of solution: 48.097167266400206

Search took: 2.2389657497406006 seconds

1.3. For maze2.world : **python astar.py world/maze2.world ASTAR**

```
Exploring neighbor (14, 88)
Exploring neighbor (15, 86)
Exploring neighbor (14, 87)
Exploring neighbor (15, 85)
Exploring neighbor (15, 87)
Exploring neighbor (16, 85)
Exploring neighbor (15, 86)
Exploring neighbor (16, 84)
Exploring neighbor (16, 86)
Exploring neighbor (17, 84)
Exploring neighbor (16, 85)
Exploring neighbor (17, 83)
Exploring neighbor (17, 85)
Exploring neighbor (17, 84)
Exploring neighbor (9, 89)
Exploring neighbor (11, 89)
Exploring neighbor (10, 88)
Exploring neighbor (10, 90)
Exploring neighbor (9, 88)
Exploring neighbor (9, 90)
Exploring neighbor (11, 90)
A* algorithm: 5487 node expansions
Quality/Cost of solution:  46.864746660604816
Search took:  2.2376110553741455 seconds
Saved image of path to file:  world/maze2_astarpath_ASTAR.png
(base) sushant@sg:~/catkin_ws/src/ai_labs$
```



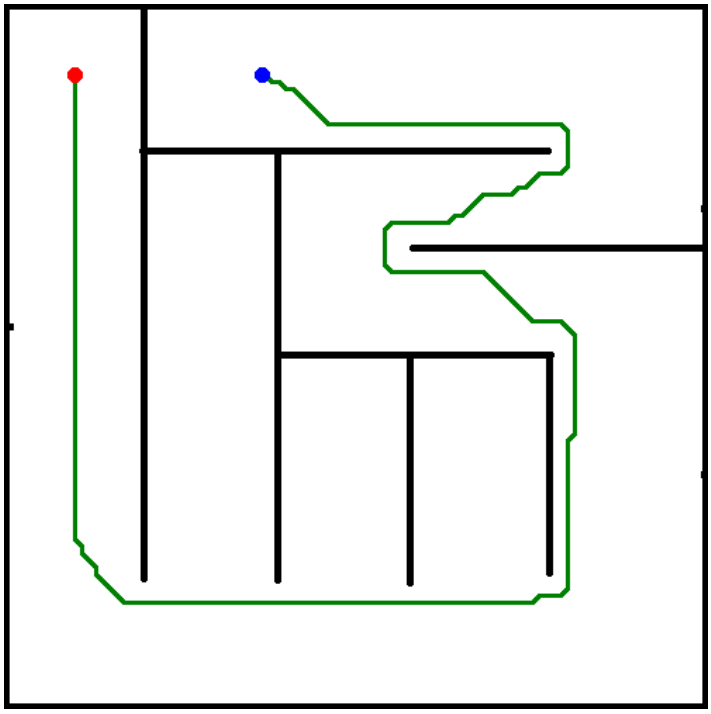
A* algorithm: 5487 node expansions

Quality/Cost of solution: 46.864746660604816

Search took: 2.2376110553741455 seconds

1.4. For maze3.world : **python astar.py world/maze3.world ASTAR**

```
Exploring neighbor (40, 14)
Exploring neighbor (39, 14)
Exploring neighbor (40, 15)
Exploring neighbor (39, 15)
Exploring neighbor (41, 15)
Exploring neighbor (40, 15)
Exploring neighbor (41, 16)
Exploring neighbor (40, 16)
Exploring neighbor (42, 16)
Exploring neighbor (41, 16)
Exploring neighbor (42, 17)
Exploring neighbor (41, 17)
Exploring neighbor (43, 17)
Exploring neighbor (42, 17)
Exploring neighbor (36, 10)
Exploring neighbor (38, 10)
Exploring neighbor (37, 9)
Exploring neighbor (37, 11)
Exploring neighbor (36, 9)
Exploring neighbor (36, 11)
Exploring neighbor (38, 9)
A* algorithm: 5772 node expansions
Quality/Cost of solution:  44.40722362970347
Search took:  2.693734884262085 seconds
Saved image of path to file:  world/maze3_astarpath_ASTAR.png
(base) sushant@sg:~/catkin_ws/src/ai_labs$
```



OBJ

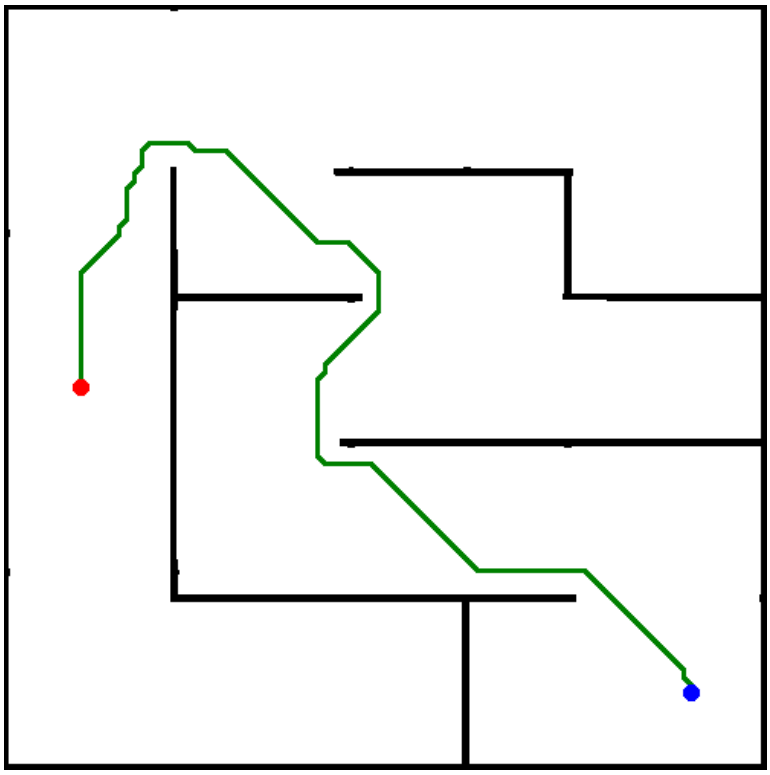
A* algorithm: 5772 node expansions

Quality/Cost of solution: 44.40722362970347

Search took: 2.693734884262085 seconds

1.5. For maze4.world : **python astar.py world/maze4.world ASTAR**

```
Exploring neighbor (85, 86)
Exploring neighbor (84, 84)
Exploring neighbor (84, 86)
Exploring neighbor (83, 84)
Exploring neighbor (84, 85)
Exploring neighbor (83, 85)
Exploring neighbor (87, 88)
Exploring neighbor (88, 89)
Exploring neighbor (87, 87)
Exploring neighbor (87, 89)
Exploring neighbor (89, 89)
Exploring neighbor (86, 87)
Exploring neighbor (87, 88)
Exploring neighbor (86, 88)
Exploring neighbor (89, 89)
Exploring neighbor (91, 89)
Exploring neighbor (90, 88)
Exploring neighbor (90, 90)
Exploring neighbor (89, 90)
Exploring neighbor (91, 88)
Exploring neighbor (91, 90)
A* algorithm: 4629 node expansions
Quality/Cost of solution:  24.438711734602883
Search took:  1.5997674465179443 seconds
Saved image of path to file:  world/maze4_astarpath_ASTAR.png
(base) sushant@sg:~/catkin_ws/src/ai_labs$
```



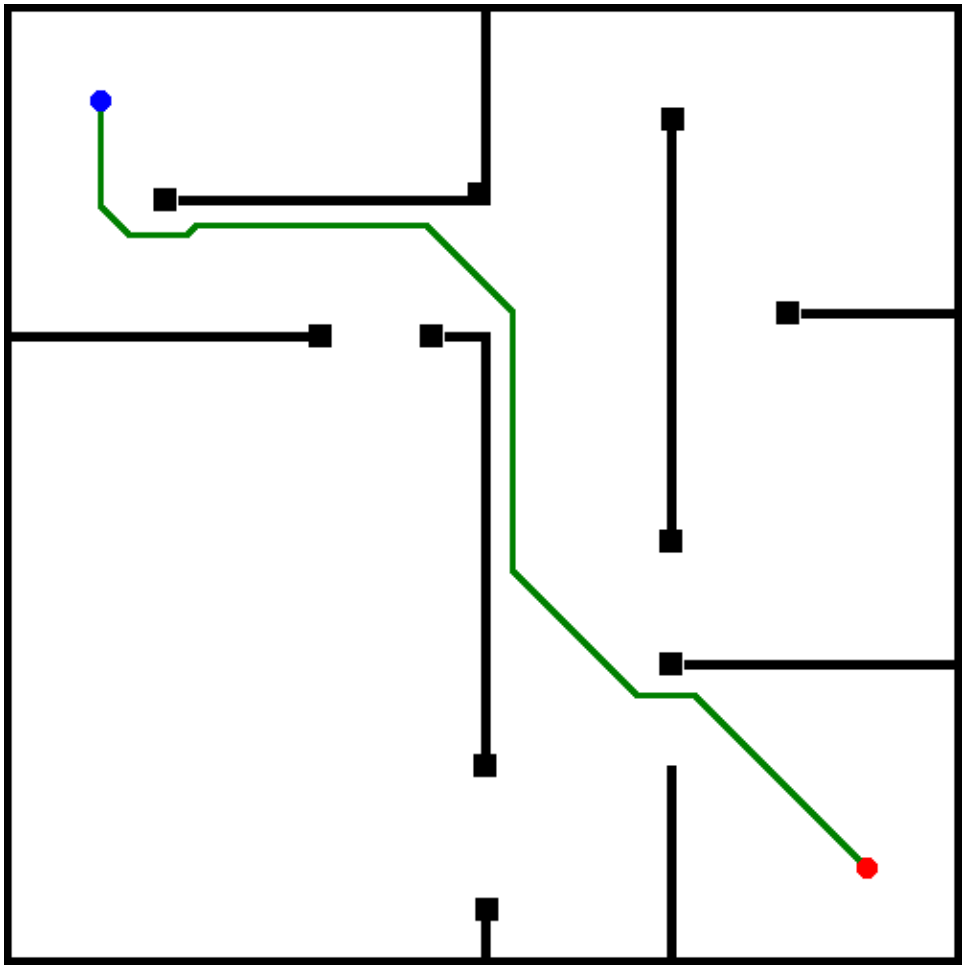
A* algorithm: 4629 node expansions

Quality/Cost of solution: 24.438711734602883

Search took: 1.5997674465179443 seconds

2. Greedy Best-first search Algorithm:

2.1. For house.world : **python astar.py world/house.world GBFS**

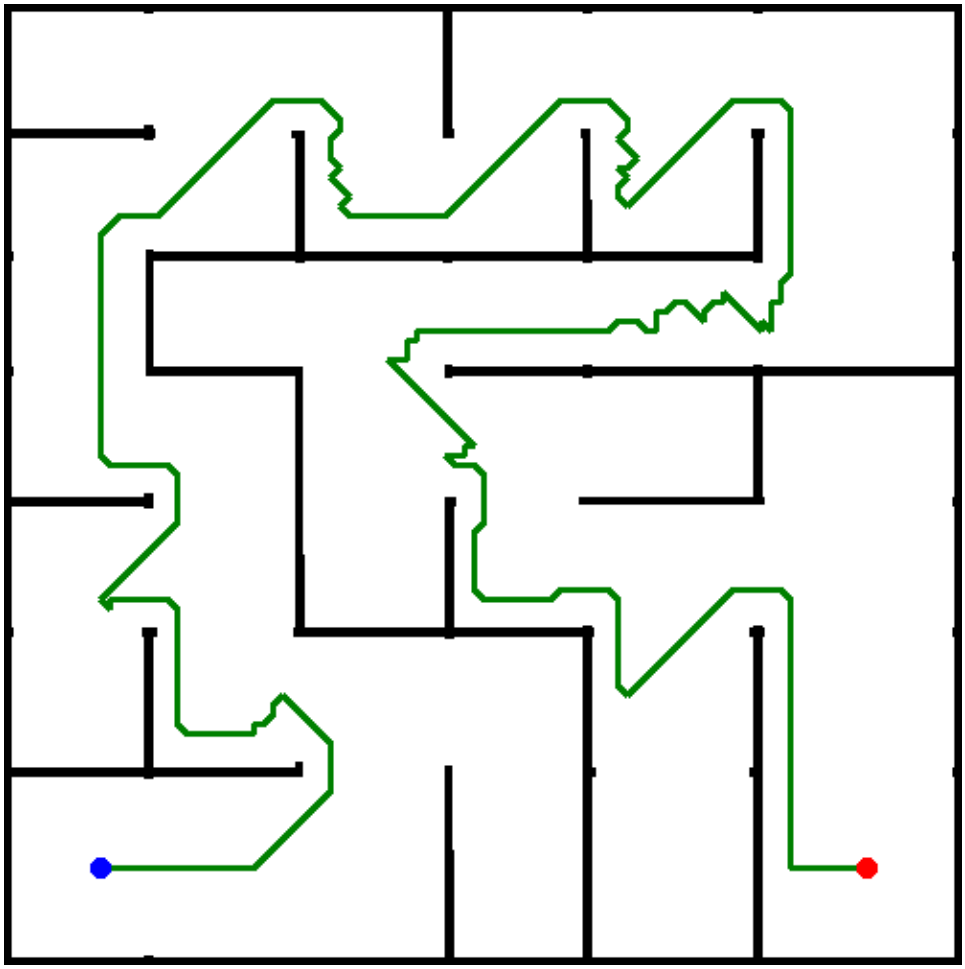


Greedy best-first search algorithm: 119 node expansions

Quality/Cost of solution: 20.41312173993557

Search took: 0.020761489868164062 seconds

2.2. For maze1.world: **python astar.py world/maze1.world GBFS**

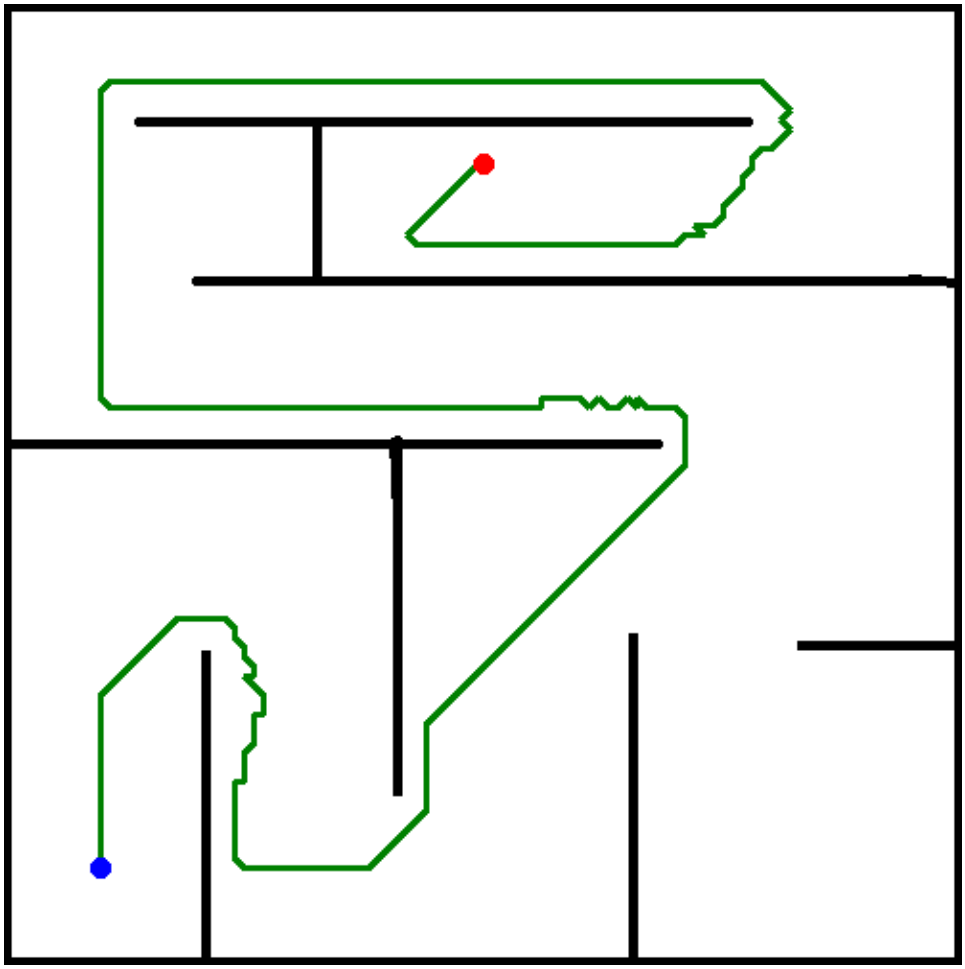


Greedy best-first search algorithm: 3371 node expansions

Quality/Cost of solution: 64.93861254635097

Search took: 1.0691170692443848 seconds

2.3. For maze2.world: **python astar.py world/maze2.world GBFS**

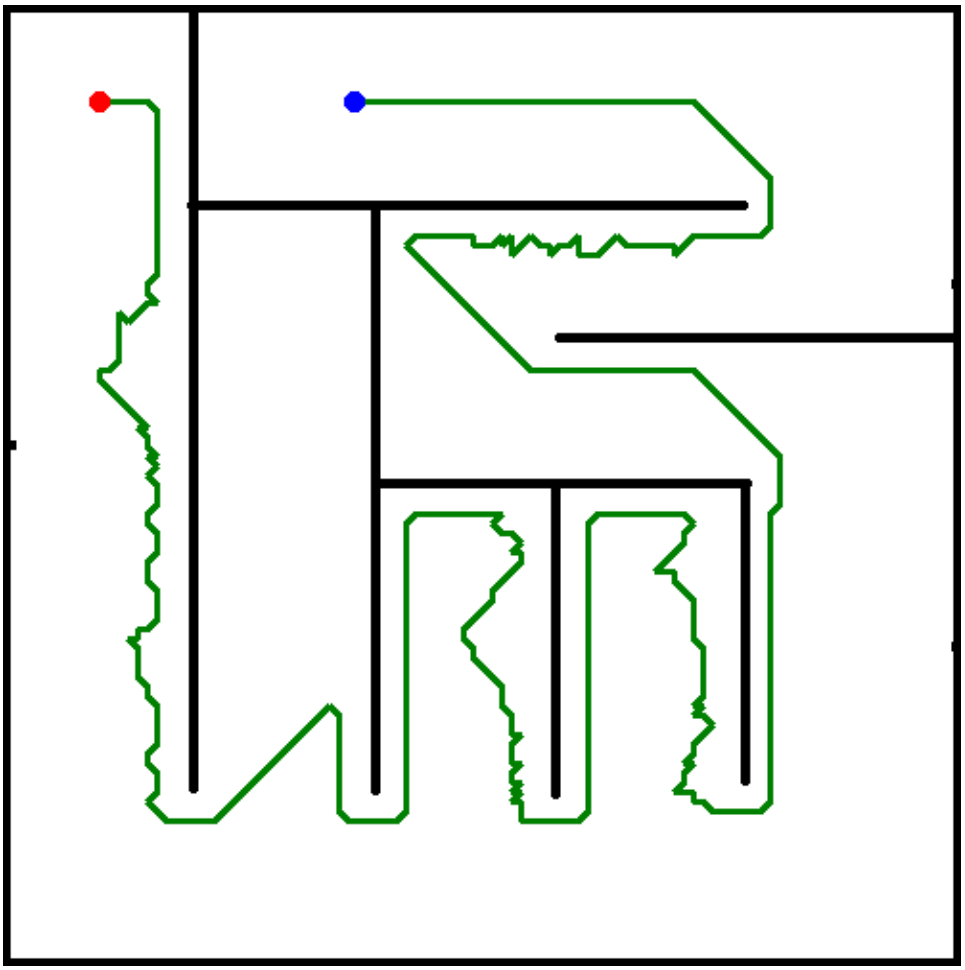


Greedy best-first search algorithm: 2244 node expansions

Quality/Cost of solution: 56.22728721360092

Search took: 0.5640408992767334 seconds

2.4. For maze3.world: **python astar.py world/maze3.world GBFS**

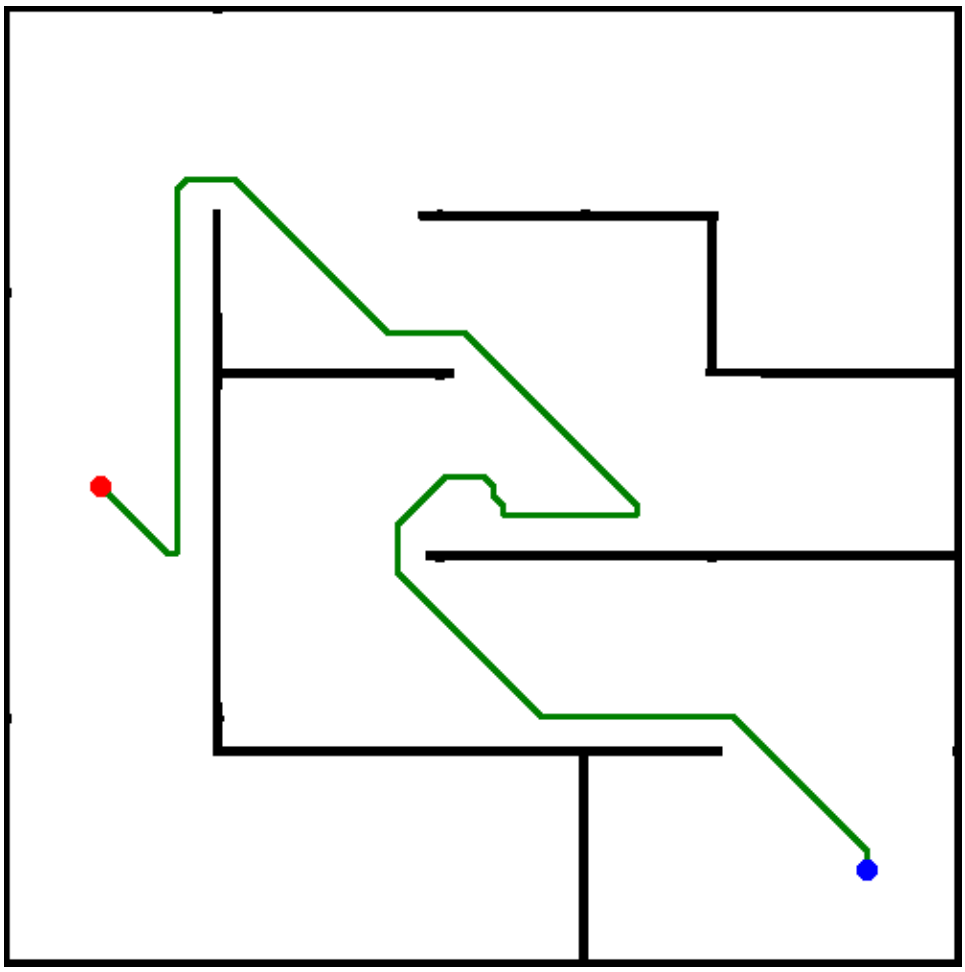


Greedy best-first search algorithm: 4128 node expansions

Quality/Cost of solution: 77.59217106341907

Search took: 1.3657448291778564 seconds

2.3. For maze4.world: **python astar.py world/maze4.world GBFS**



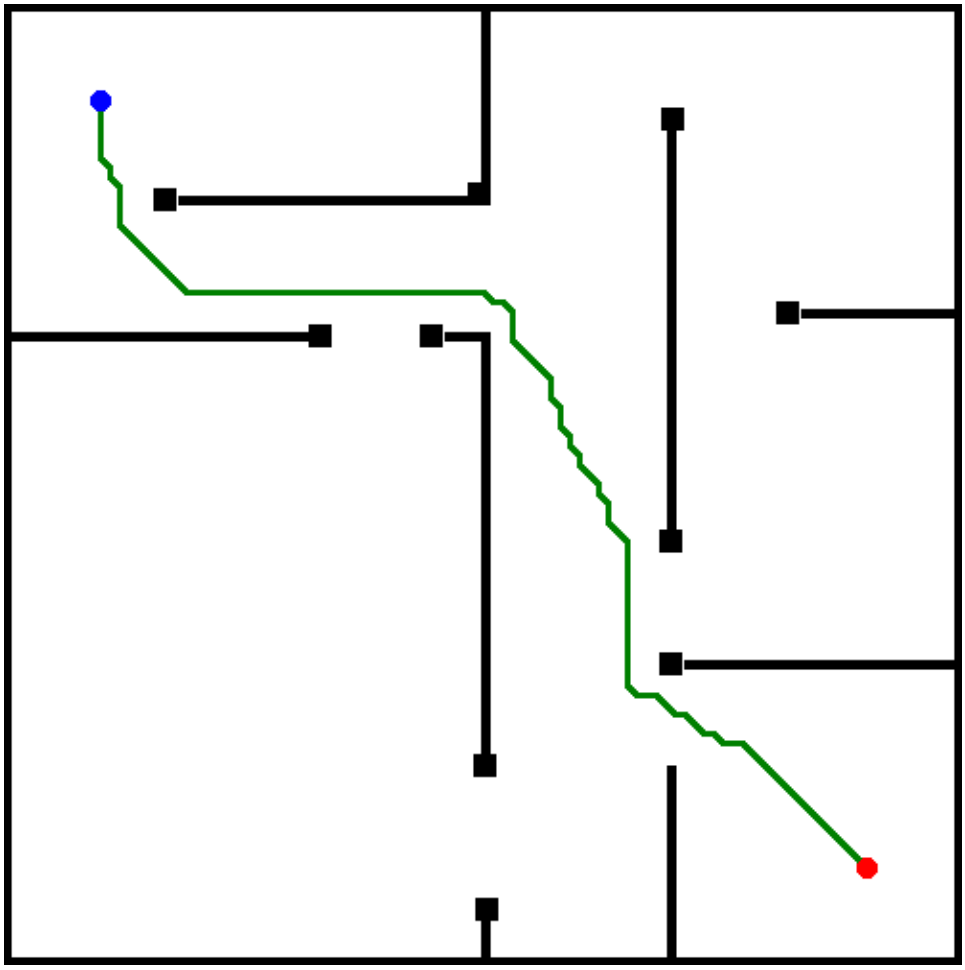
Greedy best-first search algorithm: 2521 node expansions

Quality/Cost of solution: 31.51542239853025

Search took: 0.5859439373016357 seconds

3. Greedy Best-first search Algorithm:

3.1. For house.world : **python astar.py world/house.world UCS**

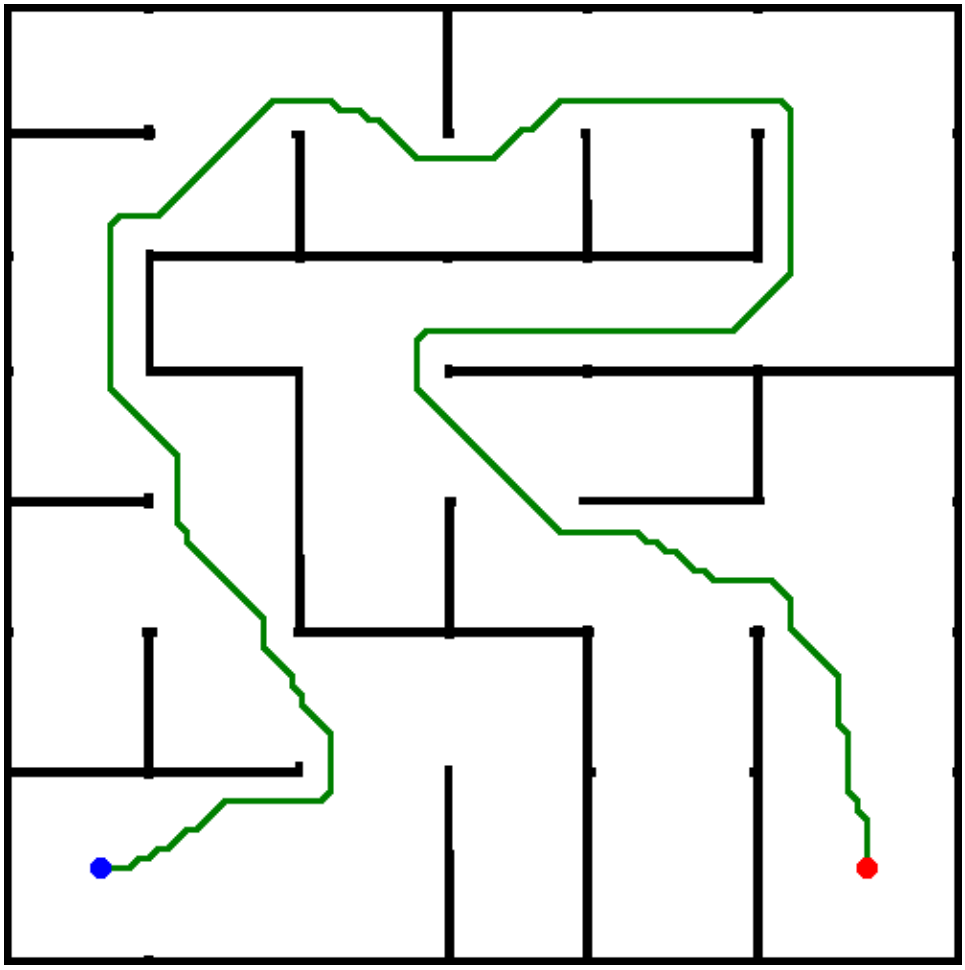


Uniform cost search algorithm: 6607 node expansions

Quality/Cost of solution: 20.289280876960824

Search took: 3.269091844558716 seconds

3.2. For maze1.world : **python astar.py world/maze1.world UCS**

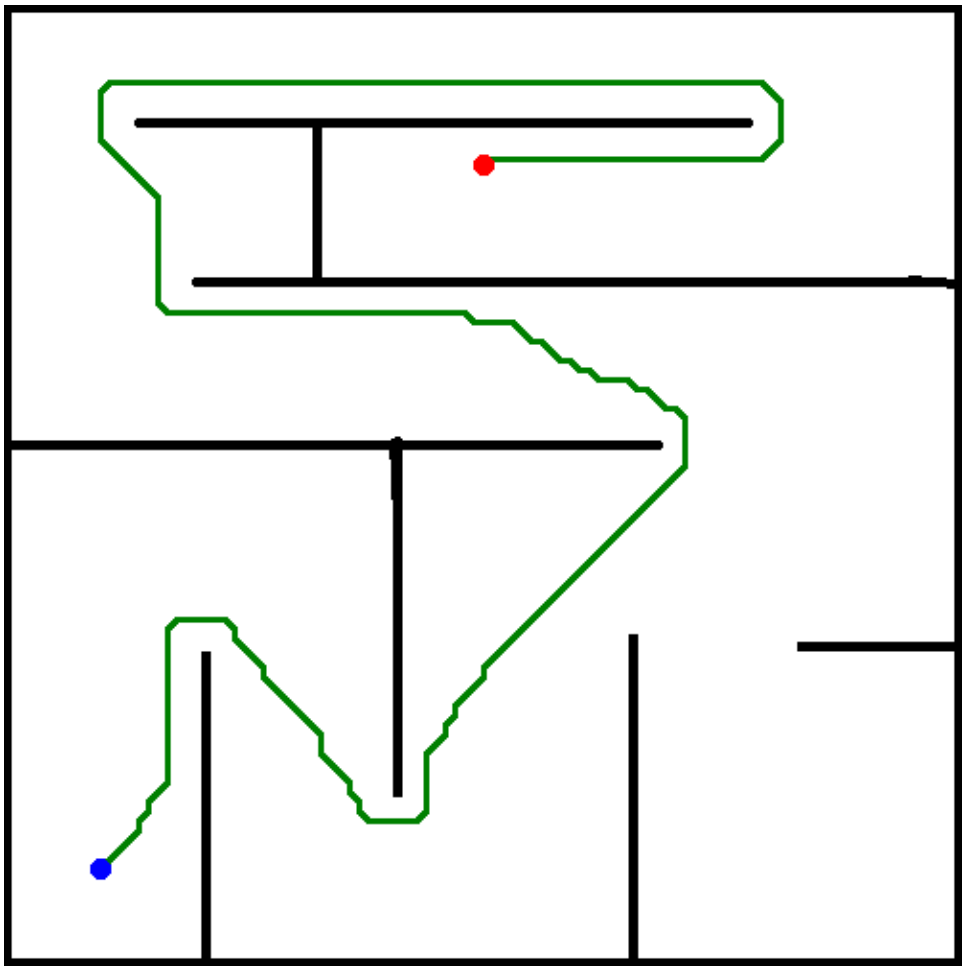


Uniform cost search algorithm: 5982 node expansions

Quality/Cost of solution: 48.09716726640021

Search took: 2.846511125564575 seconds

3.3. For maze2.world : **python astar.py world/maze2.world UCS**

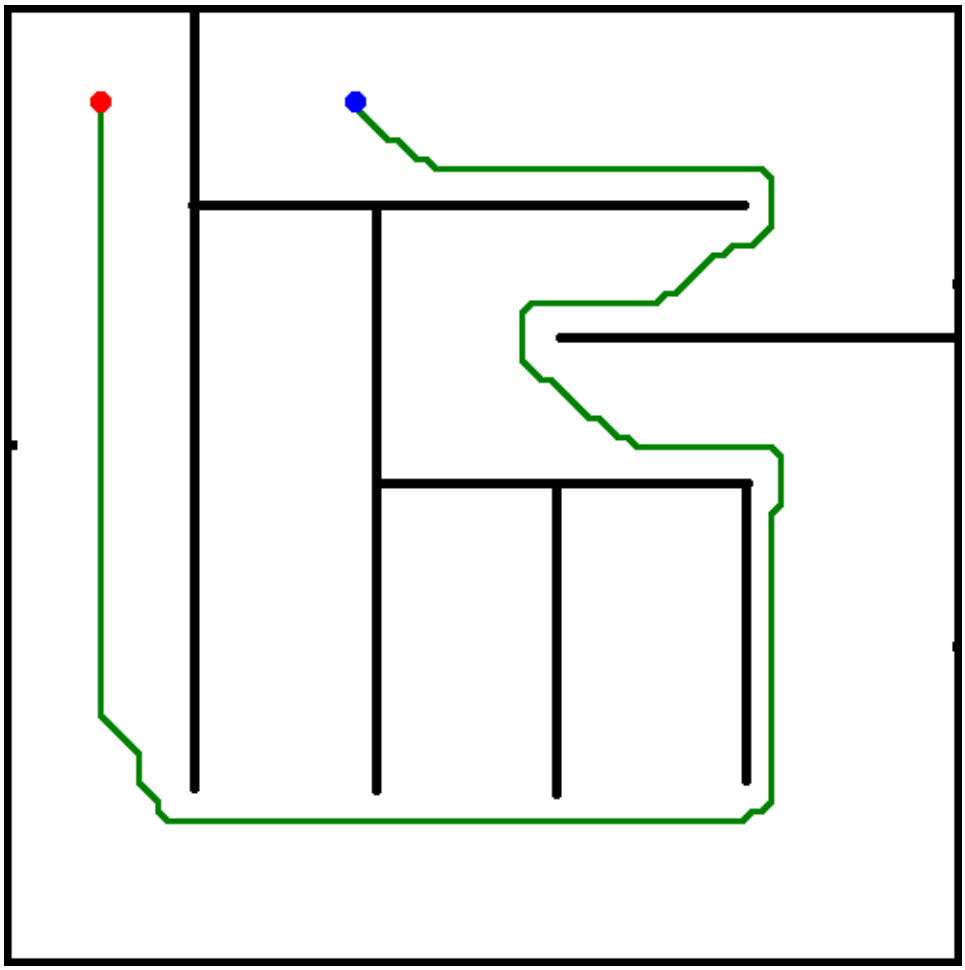


Uniform cost search algorithm: 6652 node expansions

Quality/Cost of solution: 46.864746660604816

Search took: 3.2269492149353027 seconds

3.4. For maze3.world : **python astar.py world/maze3.world UCS**

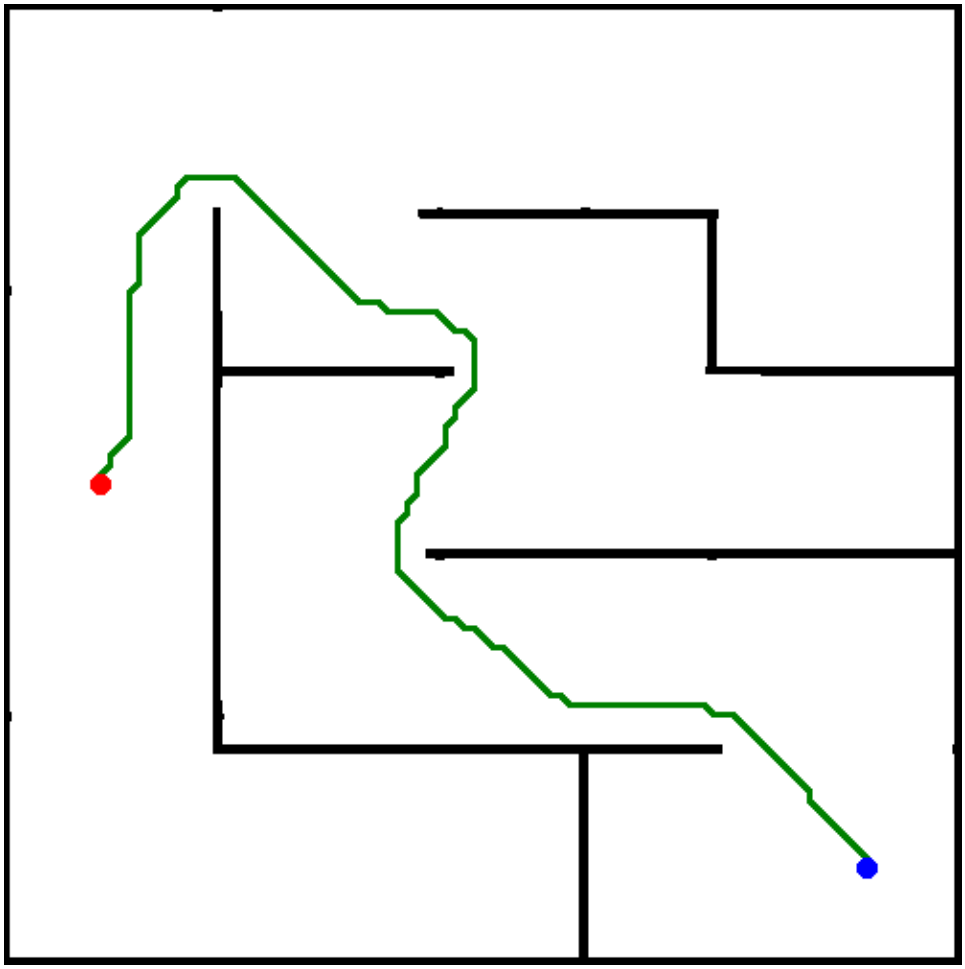


Uniform cost search algorithm: 6595 node expansions

Quality/Cost of solution: 44.40722362970349

Search took: 3.0720696449279785 seconds

3.5. For maze4.world : **python astar.py world/maze4.world UCS**



Uniform cost search algorithm: 6967 node expansions

Quality/Cost of solution: 24.438711734602872

Search took: 3.294495105743408 seconds