**2025-2\LectureScripts\Script3_HeuristicSearch\code\simulation.py**

```python
from agents import *
from environments import Maze

matrix = [[0,0,0,0,0,0,0,0],
        [0,1,0,0,1,1,1,1],
        [0,1,0,0,0,0,0,0],
        [0,1,0,0,0,0,0,0],
        [0,1,0,1,1,1,1,0],
        [0,1,0,0,0,0,0,0],
        [0,1,0,0,0,0,0,1],
        [0,1,0,0,0,0,0,0]]

'''
matrix = [[0,0,0,0],
          [1,0,1,0],
          [0,0,0,0],
          [0,1,0,1],
          [0,0,0,0]]

matrix = [[0,0,0,0],
          [0,0,0,0],
          [0,0,0,0],
          [0,0,0,0],
          [0,0,0,0],
          [0,0,0,0]]
'''

maze = Maze(matrix,[0,0],[len(matrix)-1,len(matrix[0])-1])

dfs = AgentMaze(maze, add_last, remove_last, cost_manhattan, h_manhattan)
bfs = AgentMaze(maze, add_last, remove_first, cost_manhattan, h_manhattan)
greedy= AgentMaze(maze, add_heap_greedy, remove_heap, cost_manhattan, h_manhattan)
astar = AgentMaze(maze, add_heap_astar, remove_heap, cost_manhattan, h_manhattan)

res_dfs = dfs.search()
res_bfs = bfs.search()
res_greedy = greedy.search()
res_astar = astar.search()


print(f'dfs cost:\t {res_dfs.g}')
print(f'bfs cost:\t {res_bfs.g}')
print(f'greedy cost:\t {res_greedy.g}')
print(f'astar cost:\t {res_astar.g}')


maze.plot(get_solution(res_dfs))
maze.plot(get_solution(res_bfs))
maze.plot(get_solution(res_greedy))
maze.plot(get_solution(res_astar))
```