UNIVERSITY OF ASIA PACIFIC

Department of Computer Science and Engineering

Course Title :

Artificial Intelligence and Expert Systems Lab

Course Code : CSE 404

Assignment No: 02

Submitted By:

Ripa Rani Biswas (20201001)

Md. Mominul Huque (20201049)

Anowar Hossen Farvez (20201059)

Submitted to:

Dr. Nasima Begum

Assistant Professor

Department of CSE,

University Of Asia Pacific.

Problem Title:

Intelligent Maze Navigation using A Search Algorithm.

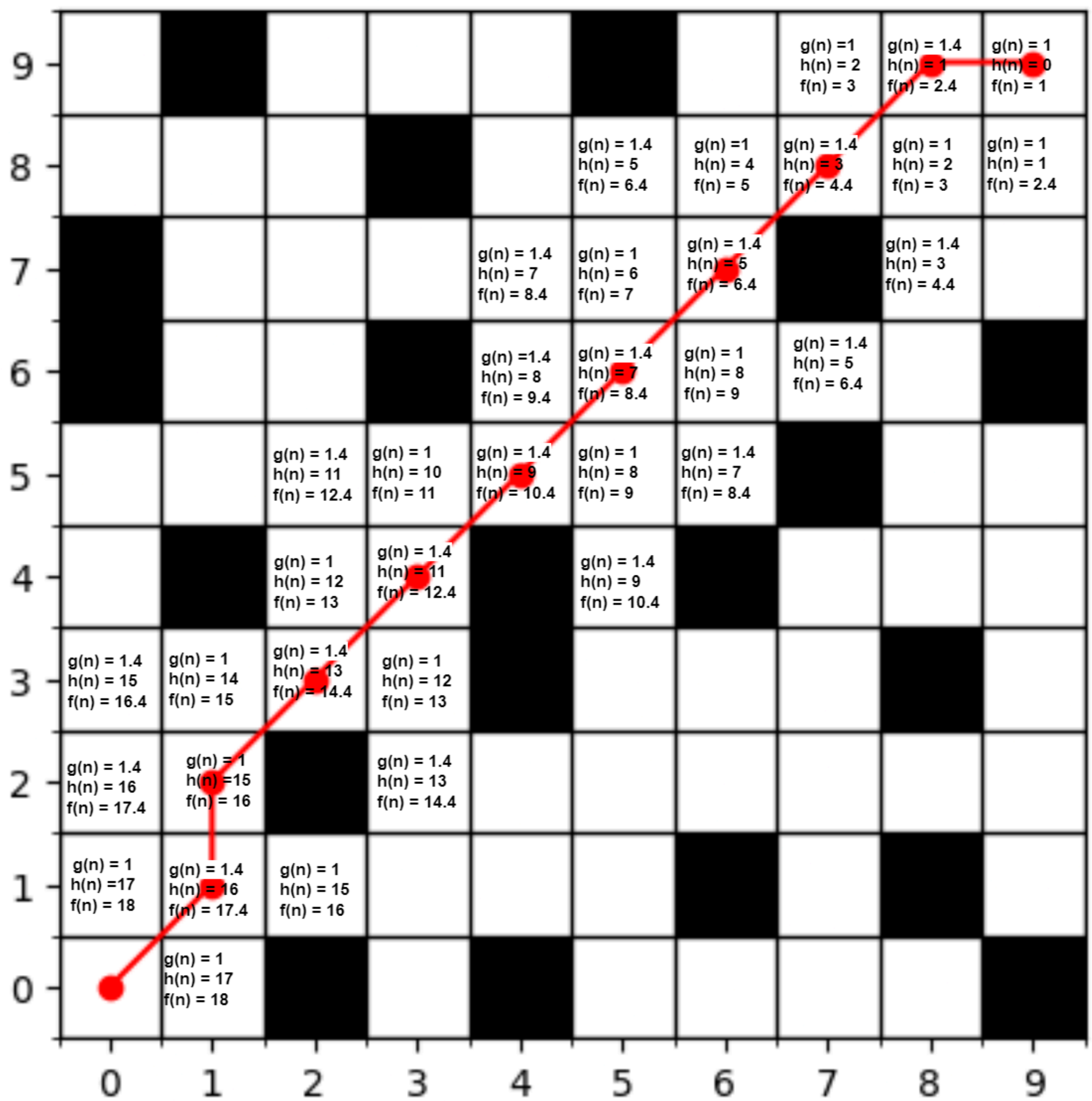
Problem Description:

The problem involves navigating through a maze represented as a grid, where each cell can either be empty or contain an obstacle. The agent is allowed to move from one cell to an adjacent cell, considering both cardinal (up, down, left, right) and diagonal directions. However, the agent cannot move to a cell occupied by an obstacle.

Tools & Languages:

- Python
- Google Colab
- Matplotlib
- Numpy

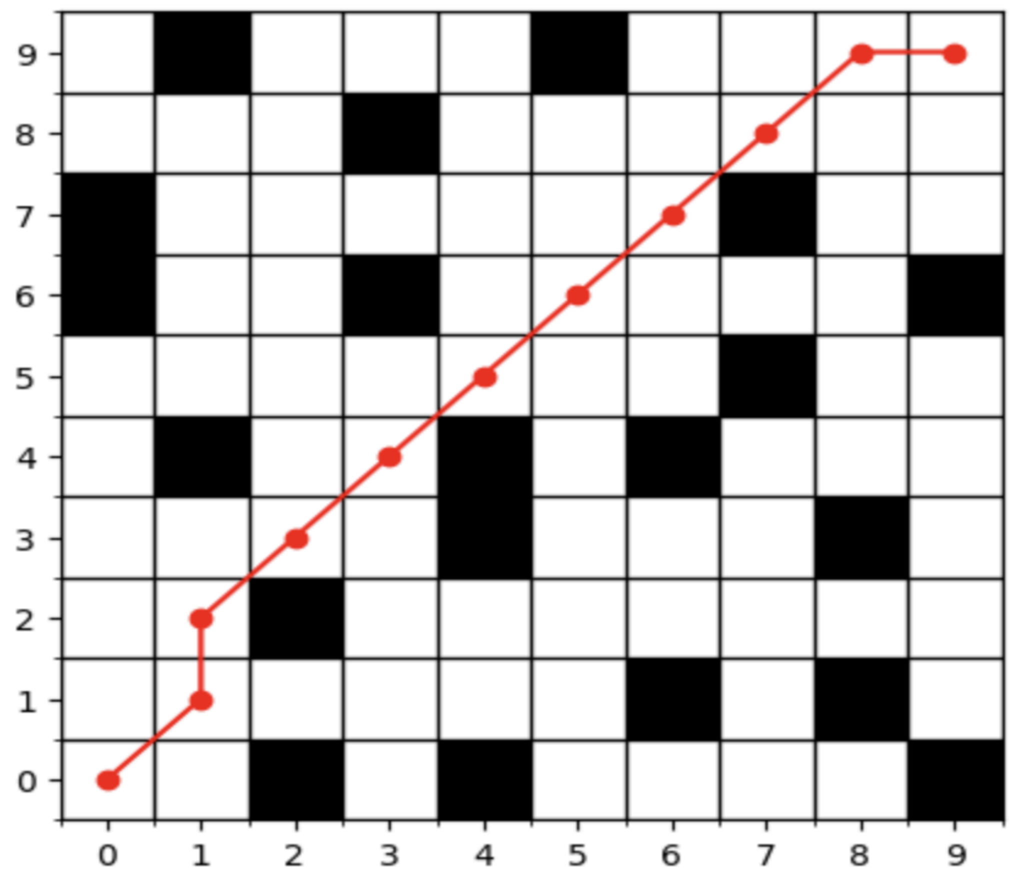
Figure:



g(n) =1
h(n) = 2
f(n) = 3
g(n) = 1.4
h(n) = 1
f(n) = 2.4
g(n) = 1
h(n) = 0
f(n) = 1
g(n) = 1.4
h(n) = 5
f(n) = 6.4
g(n) =1
h(n) = 4
f(n) = 5
g(n) = 1.4
h(n) = 3
f(n) = 4.4
g(n) = 1
h(n) = 2
f(n) = 3
g(n) = 1
h(n) = 1
f(n) = 2.4
g(n) = 1.4
h(n) = 7
f(n) = 8.4
g(n) = 1
h(n) = 6
f(n) = 7
g(n) = 1.4
h(n) = 5
f(n) = 6.4
g(n) = 1.4
h(n) = 3
f(n) = 4.4
g(n) =1.4
h(n) = 8
f(n) = 9.4
g(n) = 1.4
h(n) = 7
f(n) = 8.4
g(n) = 1
h(n) = 8
f(n) = 9
g(n) = 1.4
h(n) = 5
f(n) = 6.4
g(n) = 1.4
h(n) = 11
f(n) = 12.4
g(n) = 1
h(n) = 10
f(n) = 11
g(n) = 1.4
h(n) = 9
f(n) = 10.4
g(n) = 1
h(n) = 8
f(n) = 9
g(n) = 1.4
h(n) = 7
f(n) = 8.4
g(n) = 1
h(n) = 12
f(n) = 13
g(n) = 1.4
h(n) = 11
f(n) = 12.4
g(n) = 1.4
h(n) = 9
f(n) = 10.4
g(n) = 1.4
h(n) = 15
f(n) = 16.4
g(n) = 1
h(n) = 14
f(n) = 15
g(n) = 1.4
h(n) = 13
f(n) = 14.4
g(n) = 1
h(n) = 12
f(n) = 13
g(n) = 1.4
h(n) = 16
f(n) = 17.4
g(n) = 1
h(n) =15
f(n) = 16
g(n) = 1.4
h(n) = 13
f(n) = 14.4
g(n) = 1
h(n) =17
f(n) = 18
g(n) = 1.4
h(n) = 16
f(n) = 17.4
g(n) = 1
h(n) = 15
f(n) = 16
g(n) = 1
h(n) = 17
f(n) = 18
0 1 2 3 4 5 6 7 8 9

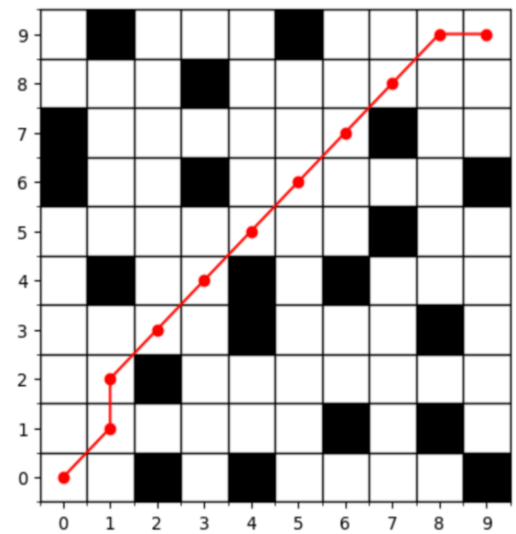
c

Solved Grid Figure-



Sample Input Output:

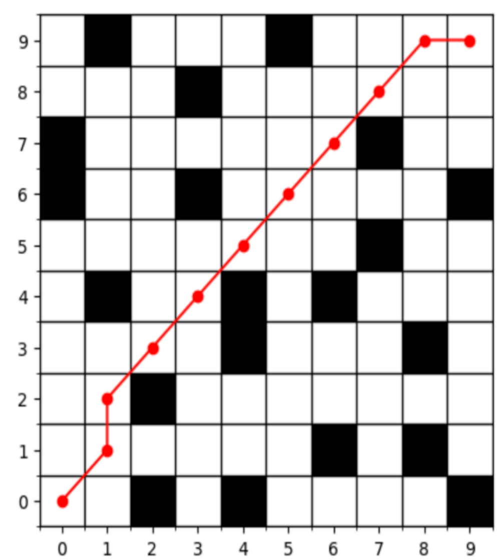
```
Evaluation Function: Euclidean + Diagonal
Optimal path - [(0, 0), (1, 1), (1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 7), (7, 8), (8, 9), (9, 9)]
Path cost: 10
Execution time: 0.001527 seconds
```



input04.txt

```
10 10
20
2 0
4 0
9 0
6 1
8 1
2 2
4 3
8 3
1 4
4 4
6 4
7 5
0 6
3 6
9 6
0 7
7 7
3 8
1 9
5 9
0 0
9 9
```

```
Evaluation Function: Euclidean + Manhattan
Optimal path - [(0, 0), (1, 1), (1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 7), (7, 8), (8, 9), (9, 9)]
Path cost: 10
Execution time: 0.000936 seconds
```



input04.txt

```
10 10
20
2 0
4 0
9 0
6 1
8 1
2 2
4 3
8 3
1 4
4 4
6 4
7 5
0 6
3 6
9 6
0 7
7 7
3 8
1 9
5 9
0 0
9 9
```

Comparison Table:

Euclidean + Manhattan Distance (Method 1)	**Euclidean + Diagonal Distance (Method 2)**
1.Path cost: 10	1.Path cost: 10
2.Execution time: 0.000936 seconds	2.Execution time: 0.001527 seconds

In this comparison, we evaluated two different heuristic methods for pathfinding: Euclidean + Manhattan Distance (Method 1) and Euclidean + Diagonal Distance (Method 2). Both methods aimed to find the optimal path through a maze from a starting point to a designated destination, while considering obstacles present in the environment.

Upon evaluation, both methods produced paths with the same cost of 10, indicating that they found equally optimal routes through the maze. However, there were slight differences in execution time between the two methods. Method 1, which employed Euclidean + Manhattan Distance, exhibited a faster execution time of approximately 0.000936 seconds, while Method 2, which utilized Euclidean + Diagonal Distance, had a slightly longer execution time of around 0.001527 seconds.

Overall, both methods successfully achieved the objective of finding an optimal path through the maze. However, Method 1 demonstrated slightly faster performance in terms of execution time.

Conclusion and Challenges:

We evaluated the A* algorithm for this problem with different heuristic algorithms and it concludes that for this problem the manhattan distance heuristic function has resulted in faster runtime.By implementing the A* search algorithm, we learned to develop a robust and efficient solution for maze navigation, demonstrating the capabilities of intelligent agents in solving complex pathfinding problems.