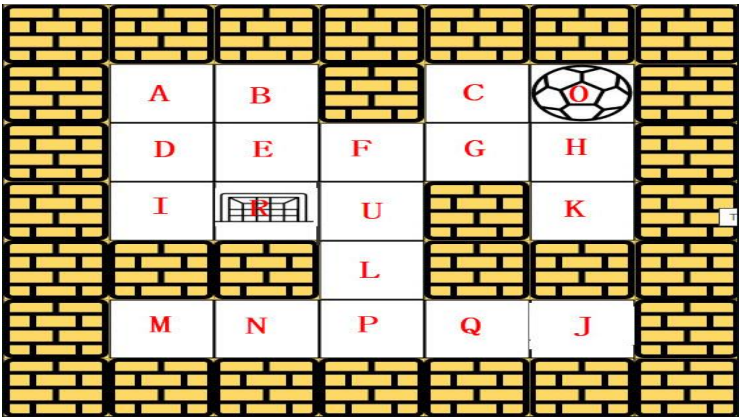
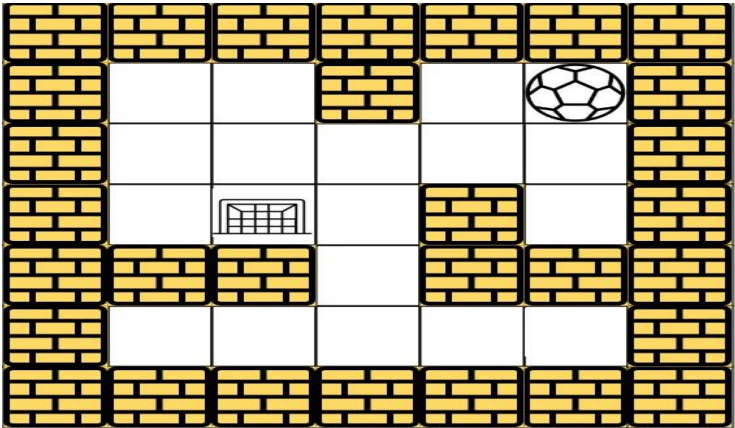


Week12 q1



Right left up down

Legged(ONE CELL AT A TIME)

	0	C	G	F	E	R
visited	0	0	0	0	0	0
queue						
print						

	0	C	G	F	E	R
visited	1	0	0	0	0	0
queue	0					
print						

	0	C	G	F	E	R
visited	1	0	0	0	0	0
queue						
print	0					

Remove 0 from queue and print 0

	0	C	G	F	E	R
visited	1	1	0	0	0	0
queue		C				
print	0					

Add C to queue and mark visited

	0	C	G	F	E	R
visited	1	1	1	0	0	0
queue			G			
print	0	C				

Add G to queue and mark G visited,remove C from queue and print C

	0	C	G	F	E	R
visited	1	1	1	1	0	0
queue				F		
print	0	C	G			

Add F to queue and mark F visited,remove G from queue and print G

	0	C	G	F	E	R
visited	1	1	1	1	1	0
queue					E	
print	0	C	G	F		

Add E to queue and mark E visited,remove F from queue and print F

	0	C	G	F	E	R
visited	1	1	1	1	1	1
queue						R
print	0	C	G	F	E	

Add R to queue and mark R visited,remove E from queue and print E

	0	C	G	F	E	R
visited	1	1	1	1	1	1
queue						R
print	0	C	G	F	E	

remove R from queue and print R

wheeled

0
visited - 0
queue-

0

visited - 1

queue- 0

add 0 to queue and mark visited

0

visited - 1

queue-

print- 0

remove 0 to queue and print 0

Visited: 0 C K

1 1 1

Queue: C K

1. Add C and K to the queue, Mark C and K as visited

Visited: 0 C K

1 1 1

Queue: K

1. Remove C from the queue

2. Print 0 C

Visited: 0 C K G

1 1 1 1

Queue: K G

1. Add G to the queue

Visited: 0 C K G

1 1 1 1

Queue: G

1. Remove K from the queue

2. Print: 0 C K

Visited: 0 C K G

1 1 1 1

Queue:

1. Remove G from the queue

2. Print 0 C K G

Visited: 0 C K G D

1 1 1 1 1

Queue: D

1. Add D to the queue

2. Mark D as visited

Visited: 0 C K G D

1 1 1 1 1

Queue:

1. Remove D from the queue

2. Print: 0 C K G D

Visited: O C K G D A I

1 1 1 1 1 1 1

Queue: A I

1. Add A, I to the queue
2. Mark A, I as visited

Visited: O C K G D A I

1 1 1 1 1 1 1

Queue: I

1. Remove A from the queue
2. Print: O C K G D A

Visited: O C K G D A I B

1 1 1 1 1 1 1 1

Queue: I B

1. Add B to the queue
2. Mark B as visited

Visited: O C K G D A I B R

1 1 1 1 1 1 1 1 1

Queue: I R

1. Add R to the queue
2. Mark R as visited

Chatgpt code

```
from collections import deque

def hasPath(maze, start, destination):
    rows, cols = len(maze), len(maze[0])
    directions = [(0, 1), (0, -1), (1, 0), (-1, 0)]

    queue = deque([start])
    visited = set()

    while queue:
        current = queue.popleft()
        if current == destination:
            return True

        visited.add(current)

        for direction in directions:
            newRow, newCol = current[0] + direction[0], current[1] + direction[1]

            while 0 <= newRow < rows and 0 <= newCol < cols and
maze[newRow][newCol] == 0:
                newRow += direction[0]
                newCol += direction[1]

            newRow -= direction[0]
            newCol -= direction[1]

            if (newRow, newCol) not in visited:
                queue.append((newRow, newCol))

    return False

# Example maze
maze = [
    [0, 0, 1, 0, 0],
    [0, 0, 0, 0, 0],
    [0, 0, 0, 1, 0],
    [1, 1, 0, 1, 1],
    [0, 0, 0, 0, 0]
]

start = (0, 4)
destination = (4,4)
print(hasPath(maze, start, destination)) # Output should be True or False
```

Test cases

File Edit Selection View Go Run ...

ALGORITHMS

week12q1.py > hasPath

```
7 queue = deque([start])
8 visited = set()
9
10 while queue:
11     current = queue.popleft()
12     if current == destination:
13         return True
14
15     visited.add(current)
16
17     for direction in directions:
18         newRow, newCol = current[0] + direction[0], current[1] + direction[1]
19
20         while 0 <= newRow < rows and 0 <= newCol < cols and maze[newRow][newCol] == 0:
21             newRow += direction[0]
22             newCol += direction[1]
23
24             newRow -= direction[0]
25             newCol -= direction[1]
26             if (newRow, newCol) not in visited:
27                 queue.append((newRow, newCol))
28
29 return False
30
31 # Example maze
32 maze = [
33     [0, 0, 1, 0, 0],
34     [1, 1, 0, 1, 1],
35     [0, 0, 0, 0, 0]
36 ]
37 start = (0, 4)
38 destination = (4, 4)
39 print(hasPath(maze, start, destination)) # Output should be True or False
40
41 maze = [
42     [0, 0, 1, 0, 0],
43     [0, 0, 0, 0, 0],
44     [0, 0, 0, 1, 0],
45     [1, 1, 0, 1, 1],
46     [0, 0, 0, 0, 0]
47 ]
48 start = (0, 4)
49 destination = (3, 2)
50 print(hasPath(maze, start, destination))
51
52 maze = [
53     [0, 0, 0, 0, 0],
54     [1, 1, 0, 0, 1],
55     [0, 0, 0, 0, 0],
56     [0, 1, 0, 0, 1],
57     [0, 1, 0, 0, 0]
58 ]
59 start = (4, 3)
60 destination = (0, 1)
61 print(hasPath(maze, start, destination))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Code

[Done] exited with code=0 in 0.178 seconds

[Running] python -u "c:\Users\cheth\OneDrive\Desktop\SFBUS\SEM3\ALGORITHMS\week12q1.py"

True

False

False

[Done] exited with code=0 in 0.203 seconds

Activate Windows
Go to Settings to activate Windows.

73°F Sunny

Ln 22, Col 39 Spaces: 4 UTF-8 CRLF Python 3.9.13 64-bit (microsoft store)

12:19 PM 8/8/2023

File Edit Selection View Go Run ...

ALGORITHMS

week12q1.py > ...

```
33 [0, 0, 0, 1, 0],
34 [1, 1, 0, 1, 1],
35 [0, 0, 0, 0, 0]
36 ]
37 start = (0, 4)
38 destination = (4, 4)
39 print(hasPath(maze, start, destination)) # Output should be True or False
40
41 maze = [
42     [0, 0, 1, 0, 0],
43     [0, 0, 0, 0, 0],
44     [0, 0, 0, 1, 0],
45     [1, 1, 0, 1, 1],
46     [0, 0, 0, 0, 0]
47 ]
48 start = (0, 4)
49 destination = (3, 2)
50 print(hasPath(maze, start, destination))
51
52 maze = [
53     [0, 0, 0, 0, 0],
54     [1, 1, 0, 0, 1],
55     [0, 0, 0, 0, 0],
56     [0, 1, 0, 0, 1],
57     [0, 1, 0, 0, 0]
58 ]
59 start = (4, 3)
60 destination = (0, 1)
61 print(hasPath(maze, start, destination))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Code

[Done] exited with code=0 in 0.178 seconds

[Running] python -u "c:\Users\cheth\OneDrive\Desktop\SFBUS\SEM3\ALGORITHMS\week12q1.py"

True

False

False

Activate Windows
Go to Settings to activate Windows.

74°F Sunny

Ln 56, Col 2 Spaces: 4 UTF-8 CRLF Python 3.9.13 64-bit (microsoft store)

12:21 PM 8/8/2023

