Green University of Bangladesh
Department of Computer Science and Engineering(CSE)

Faculty of Science and Engineering

Semester: (Spring, Year:2025), B.Sc. in CSE (Day)

Lab Report 01

Course Title: Artificial Intelligence Lab

Course Code: CSE 316 **Section: 221 D9**

Lab Experiment Name: BFS Traversal on a 2D Grid with Path Reconstruction

Student Details

Name	ID
Shahadat Hosen Nishan	221002099

Lab Date	:	**02/02/2025**
Submission Date	:	**26/04/2025**
Course Teacher's Name	:	**Md. Riad Hassan**

[For Teachers use only: Don't Write Anything inside this box]

Lab Report

Status Marks: …………………………………… **Signature:…………………**

Comments:……………………………………… **Date:…………………………**

Title:

BFS Traversal on a 2D Grid with Path Reconstruction

Objective:

The objective of this lab is to implement the Breadth-First Search (BFS) algorithm to navigate a robot from a start point to a destination point on a 2D grid, and to print the shortest path traversed by the robot. The solution also involves saving parent state information during traversal and using that information to reconstruct and print the path recursively.

Problem Statement:

We are tasked with simulating the movement of a robot on a 2D grid. The robot starts at a specific start position and aims to reach a destination position. We are required to implement the BFS algorithm to explore the grid, ensuring the shortest path is found. The challenge includes:

1. Implementing BFS to explore all possible moves.
2. Tracking each cell's parent state (i.e., where it came from).
3. Using parent information to reconstruct and print the path from the start point to the destination.

Algorithm:

Breadth-First Search (BFS) Algorithm:

Steps:

1. Initialization:
    a. Define the grid with some obstacles.
    b. Define the start and destination points.
2. BFS Traversal:
    a. Start from the source node and explore all possible neighbors (up, down, left, right).
    b. Mark visited nodes and stored parent information for path reconstruction.
    c. If the destination node is reached, stop the search.
3. Path Reconstruction:
    a. Using the parent state information, recursively trace back from the destination to the start, printing the path.

Time Complexity:

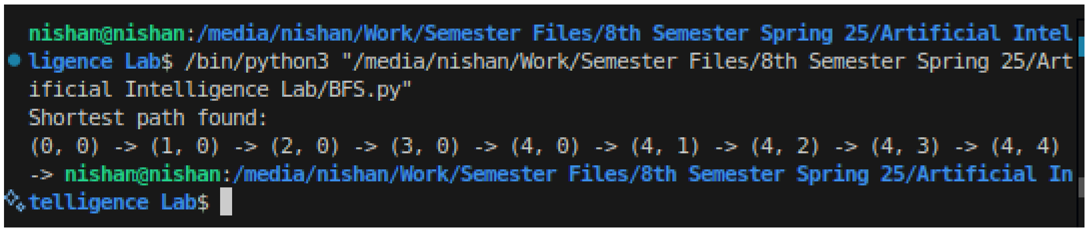
Time Complexity: O(V + E)
Space Complexity: O(V)

Implementation:

Code:

```
from collections import deque

class Node:
    def __init__(self, x, y, parent=None):
        self.x = x
        self.y = y
        self.parent = parent

def bfs(grid, start, goal):

          directions = [(-1, 0), (1, 0), (0, -1), (0, 1)]

          queue = deque([Node(start[0], start[1])])

          visited = set()
          visited.add((start[0], start[1]))

          while queue:
              current = queue.popleft()

              if (current.x, current.y) == goal:
                  return current

              for direction in directions:
                  new_x, new_y = current.x + direction[0], current.y + direction[1]

                  if 0 <= new_x < len(grid) and 0 <= new_y < len(grid[0]) and grid[new_x][new_y] == 0 and (new_x, new_y) not in visited:
                      visited.add((new_x, new_y))
                      queue.append(Node(new_x, new_y, current))

          return None

      def print_path(node):
          if node is None:
              return
          print_path(node.parent)
          print(f"({node.x}, {node.y})", end=" -> ")

      if __name__ == "__main__":
          grid = [
              [0, 0, 0, 1, 0],
              [0, 1, 0, 1, 0],
              [0, 1, 0, 0, 0],
              [0, 0, 0, 1, 0],
              [0, 0, 0, 0, 0]
          ]

          start = (0, 0)
          goal = (4, 4)

          destination_node = bfs(grid, start, goal)

          if destination_node is None:
              print("No path found!")
          else:
              print("Shortest path found:")
              print_path(destination_node)

```

Output:



```
nishan@nishan:/media/nishan/Work/Semester Files/8th Semester Spring 25/Artificial Intel
ligence Lab$ /bin/python3 "/media/nishan/Work/Semester Files/8th Semester Spring 25/Art
ificial Intelligence Lab/BFS.py"
Shortest path found:
(0, 0) -> (1, 0) -> (2, 0) -> (3, 0) -> (4, 0) -> (4, 1) -> (4, 2) -> (4, 3) -> (4, 4)
-> nishan@nishan:/media/nishan/Work/Semester Files/8th Semester Spring 25/Artificial In
telligence Lab$ 
```

fig: Implementing BFS Traversal on a 2D Grid

Conclusion:

In this lab, we successfully implemented the Breadth-First Search (BFS) algorithm to traverse a robot from a start position to a destination on a 2D grid. The robot's movement was simulated, and the path taken was reconstructed and printed using parent state information.

This exercise demonstrated how BFS ensures the shortest path in an unweighted grid environment, and the concept of path reconstruction using parent pointers was effectively used to trace the path.