



**Green University of Bangladesh**  
**Department of Computer Science and Engineering (CSE)**  
**Faculty of Sciences and Engineering**  
**Semester: (Spring, Year:2023), BSc. in CSE (Day)**

**LAB REPORT - 02**

**Course Title:** Artificial Intelligence Lab

**Course Code:** CSE-316

**Section:** PC-201 DC

**Student Details**

Name		Students Id
1.	Md. Romzan Alom	201902144

**Lab Date:** 29-04-2023

**Submission Date:** 05-05-2023

**Course Teacher's Name:** Rusmita Halim Chaity

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

**Lab Report Status**

**Marks:** .....

**Signature:** .....

**Comments:** .....

**Date:** .....

## 1. TITLE OF THE LAB EXPERIMENT

Implementation Breadth First Search (BFS) using python.

## 2. OBJECTIVES/AIM

- To learn about Breadth First Search.
- To acquire knowledge about python.
- To implement BFS on python.
- To learn python operators.
- To learn conditional statements in python.
- To learn loops in python.
- To learn recursive functions in python.

## 3. PROCEDURE / ANALYSIS / DESIGN

**Problem:** Write a program where a robot traverses on a 2D plane using the BFS algorithm and goes from start to destination point. Now print the path it will traverse to reach the destination.

(Hint: Save parent state information and use that information to write a recursive function for printing the path.)

## 4. IMPLEMENTATION

### Code of this problem:

```
ROWS = int(input("Enter the value of rows: "))
COLS = int(input("Enter the value of coloums: "))
grid = [[0 for j in range(COLS)] for i in range(ROWS)]
for i in range(ROWS):
    for j in range(COLS):
        grid[i][j] = int(input("Enter the value for position ({}, {}): ".format(i, j)))
print("The 2D array is:")
for i in range(ROWS):
    for j in range(COLS):
        print(grid[i][j], end=' ')
    print()
def is_valid(x, y):
    return x >= 0 and x < ROWS and y >= 0 and y < COLS and grid[x][y] == 0

def bfs(start, dest, visited, path):
    if start in visited:
        return None
    visited.add(start)
    path.append(start)
    if start == dest:
```

```

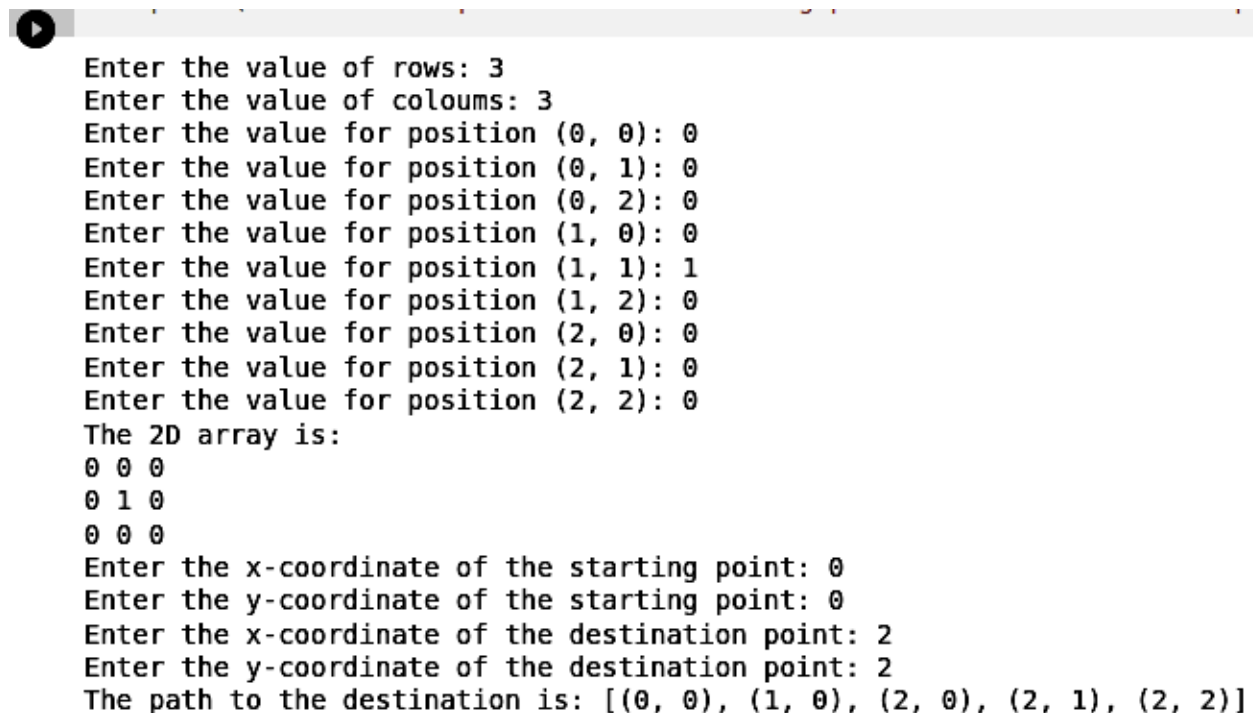
return path
x, y = start
neighbors = [(x+1, y), (x-1, y), (x, y+1), (x, y-1)]
for neighbor in neighbors:
    if is_valid(*neighbor):
        result = bfs(neighbor, dest, visited, path)
        if result is not None:
            return result
path.pop()
return None

start_x = int(input("Enter the x-coordinate of the starting point: "))
start_y = int(input("Enter the y-coordinate of the starting point: "))
start = (start_x, start_y)
dest_x = int(input("Enter the x-coordinate of the destination point: "))
dest_y = int(input("Enter the y-coordinate of the destination point: "))
dest = (dest_x, dest_y)

visited = set()
path = []
result = bfs(start, dest, visited, path)
if result is not None:
    print("The path to the destination is:", result)
else:
    print("There is no path from the starting point to the destination point.")

```

## 5. TEST RESULT / OUTPUT



```

Enter the value of rows: 3
Enter the value of coloums: 3
Enter the value for position (0, 0): 0
Enter the value for position (0, 1): 0
Enter the value for position (0, 2): 0
Enter the value for position (1, 0): 0
Enter the value for position (1, 1): 1
Enter the value for position (1, 2): 0
Enter the value for position (2, 0): 0
Enter the value for position (2, 1): 0
Enter the value for position (2, 2): 0
The 2D array is:
0 0 0
0 1 0
0 0 0
Enter the x-coordinate of the starting point: 0
Enter the y-coordinate of the starting point: 0
Enter the x-coordinate of the destination point: 2
Enter the y-coordinate of the destination point: 2
The path to the destination is: [(0, 0), (1, 0), (2, 0), (2, 1), (2, 2)]

```

**Figure\_01:** Output of this problem

In this problem, the user inputs a 3x3 2D array with a starting point of (0, 0) and a destination point of (2, 2). The program then uses a recursive BFS algorithm to find the shortest path from the starting point to the destination point in the array. The resulting path is printed as a list of tuples, where each tuple represents a point in the path. In this case, the shortest path goes through the points (0, 0), (1, 0), (2, 0), (2, 1), and (2, 2).

## **6. ANALYSIS AND DISCUSSION**

This experiment mainly based on software. So, it may have so software error. Based on the focused objective(s) to learn the step-by-step working system of that problem. This problem is similar like BFS that's why I can easily understand. The task will help us to create new new solution from different problems. The main hard part of this experiment is successfully completed traversal. We face so many problem for understanding recursive function of python and their working system. Now, we get so many knowledge to execute those function. Those are very important for our future lab task.