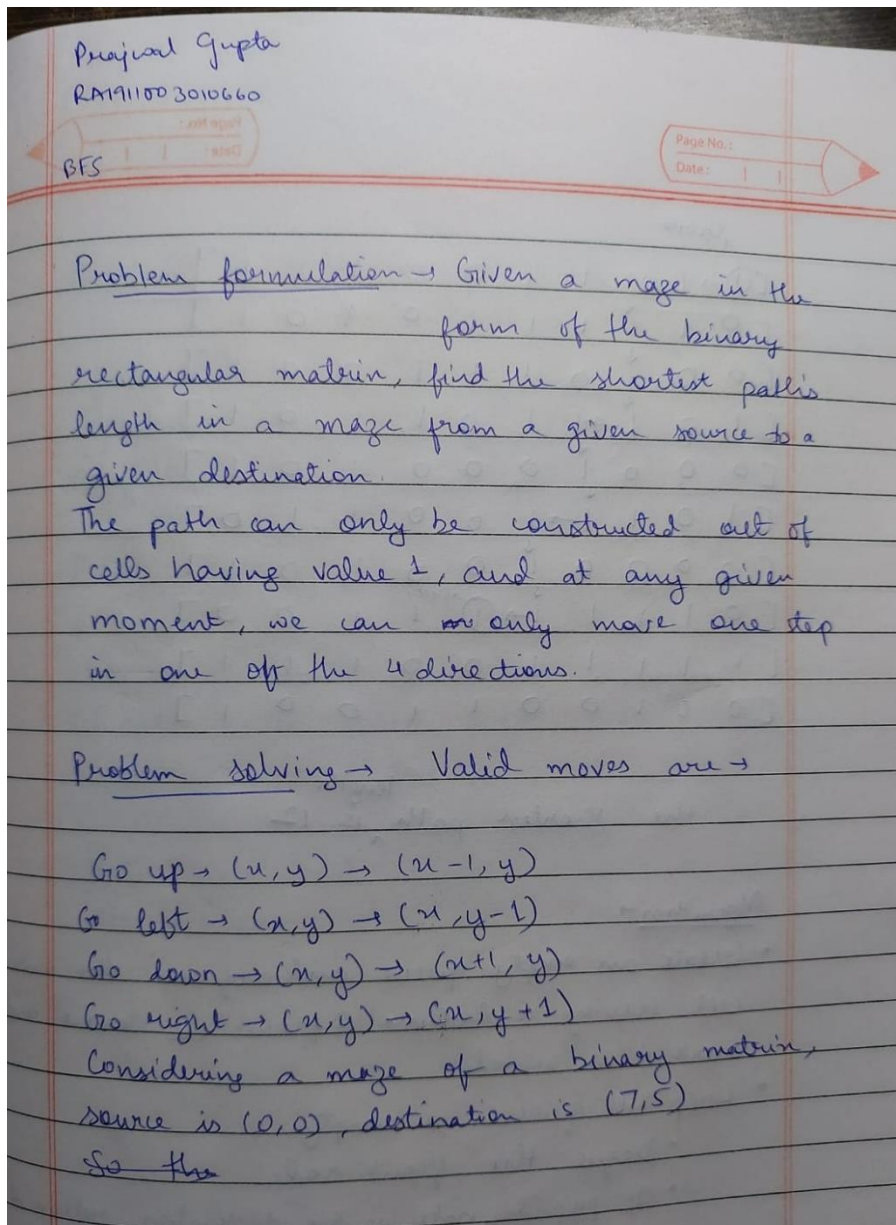


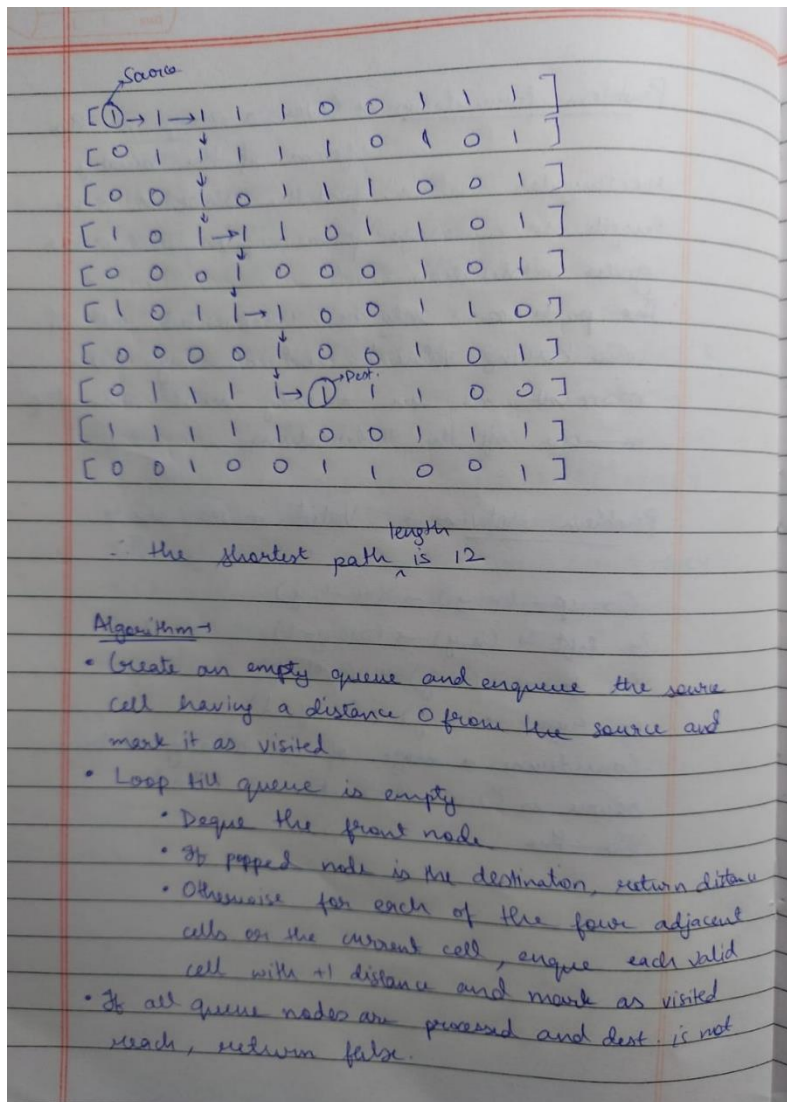
Prajwal Gupta

RA1911003010660

AI LAB 4(A)

Aim-Implementation and Analysis of BFS for an application (Shortest Path in a maze)





Code-

```
import sys
```

```
from collections import deque
```

```
row = [-1, 0, 0, 1]
```

```
col = [0, -1, 1, 0]
```

```
def isValid(mat, visited, row, col):
```

```
    return (row >= 0) and (row < len(mat)) and (col >= 0) and (col < len(mat[0])) \
           and mat[row][col] == 1 and not visited[row][col]
```

```
def findShortestPathLength(mat, src, dest):

    i, j = src

    x, y = dest

    if not mat or len(mat) == 0 or mat[i][j] == 0 or mat[x][y] == 0:
        return -1

    (M, N) = (len(mat), len(mat[0]))

    visited = [[False for x in range(N)] for y in range(M)]

    q = deque()

    visited[i][j] = True

    q.append((i, j, 0))

    min_dist = sys.maxsize

    while q:

        (i, j, dist) = q.popleft()

        if i == x and j == y:
            min_dist = dist
            break

        for k in range(4):
```

```

        if isValid(mat, visited, i + row[k], j + col[k]):
            visited[i + row[k]][j + col[k]] = True
            q.append((i + row[k], j + col[k], dist + 1))

    if min_dist != sys.maxsize:
        return min_dist
    else:
        return -1

if __name__ == '__main__':

    mat = [
        [1, 1, 1, 1, 1, 0, 0, 1, 1, 1],
        [0, 1, 1, 1, 1, 1, 0, 1, 0, 1],
        [0, 0, 1, 0, 1, 1, 1, 0, 0, 1],
        [1, 0, 1, 1, 1, 0, 1, 1, 0, 1],
        [0, 0, 0, 1, 0, 0, 0, 1, 0, 1],
        [1, 0, 1, 1, 1, 0, 0, 1, 1, 0],
        [0, 0, 0, 0, 1, 0, 0, 1, 0, 1],
        [0, 1, 1, 1, 1, 1, 1, 1, 0, 0],
        [1, 1, 1, 1, 1, 0, 0, 1, 1, 1],
        [0, 0, 1, 0, 0, 1, 1, 0, 0, 1]
    ]

    src = (0, 0)
    dest = (7, 5)

    min_dist = findShortestPathLength(mat, src, dest)

    if min_dist != -1:

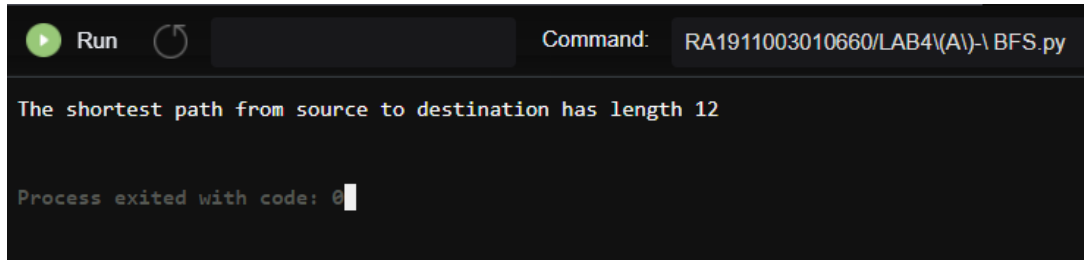
```

```
print("The shortest path from source to destination has length", min_dist)

else:

    print("Destination cannot be reached from source")
```

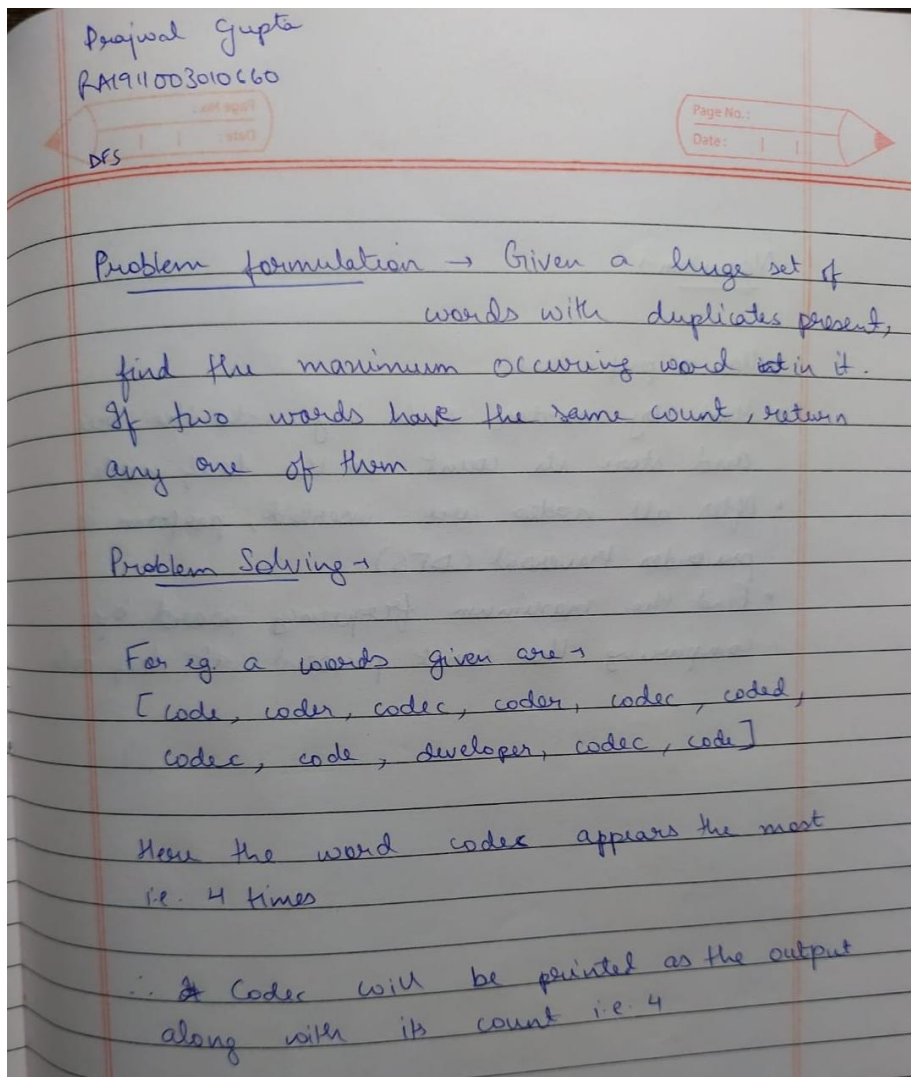
Output-



The screenshot shows a code execution interface with a dark background. At the top, there is a 'Run' button with a green play icon and a refresh icon. To the right, the command 'RA1911003010660/LAB4\AI-1 BFS.py' is displayed. The main output area shows the text 'The shortest path from source to destination has length 12'. Below this, it says 'Process exited with code: 0'.

AI LAB 4(B)

Aim- Implementation and Analysis of DFS for an application (Maximum occurring word)



Algorithm

- Use a prefix tree
- Start by inserting each key into the tree and store its count in the leaf nodes
- After all nodes are inserted, perform its preorder traversal (DFS)
- Find the maximum frequency word by comparing the count present at leaf nodes.

Code-

```
class TrieNode:
```

```
    def __init__(self):
```

```
        self.key = None
```

```
        self.count = 0
```

```
        self.character = {}
```

```
#string in tree
```

```
def insert(head, s):
```

```
    curr = head
```

```
    for c in s:
```

```
        curr = curr.character.setdefault(c, TrieNode())
```

```
    curr.key = s
```

```
    curr.count += 1
```

```
def preorder(curr, key="", max_count=0):
```

```
    if curr is None:
```

```
        return key, max_count
```

```

for (k, v) in curr.character.items():

    if max_count < v.count:

        key = v.key

        max_count = v.count

    key, max_count = preorder(v, key, max_count)

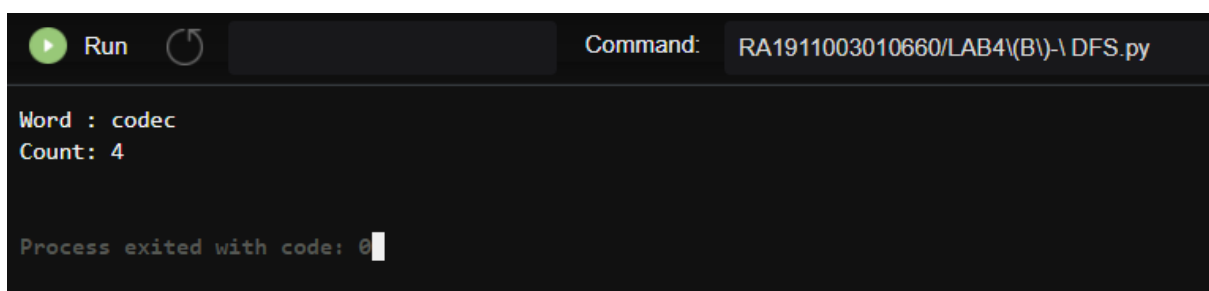
return key, max_count

if __name__ == '__main__':
    words = [
        'code', 'coder', 'coding', 'codable', 'codec', 'codecs', 'coded',
        'codeless', 'codec', 'codecs', 'codependence', 'codex', 'codify',
        'codependents', 'codes', 'code', 'coder', 'codesign', 'codec',
        'codeveloper', 'codrive', 'codec', 'codecs', 'codiscovered'
    ]

    head = TrieNode()
    for word in words:
        insert(head, word)
    key, count = preorder(head)
    print('Word :', key)
    print('Count:', count)

```

Output-



```

Run Command: RA1911003010660/LAB4\B\ DFS.py

Word : codec
Count: 4

Process exited with code: 0

```