# LAB 1 REPORT

## CS312

*Soumya Srividhya, 170010038*
*Anudeep Tubati, 170010039*

## INTRODUCTION

This lab assignment simulates the state space search using 3 uninformed searching algorithms to help Pacman solve the maze it is in and reach its goal. The 3 algorithms used are; Breadth-First Search, Depth-First Search and Depth-First Iterative Deepening. This report elaborates on the pseudo-code, results, dependency of the results on the order of neighbours added and their conclusions.

## PSEUDO-CODE

I. **MoveGen(N)**

```
nextStates <= ()
for neighbor n of state N in order (DOWN, UP, RIGHT, LEFT) do
        if n is not boundary and n not in (open ∪ closed) then
                nextStates.append(n)
return nextStates
```

II. **GoalTest(N)**

```
if N.data == '*' then
        return true
else
        return false
```

## RESULTS

Down > Up > Right > Left

| Maze Size | Number of States Explored | | | Length of the Path Found | | |
|---|---|---|---|---|---|---|
| Algorithm | BFS | DFS | DFID | BFS | DFS | DFID |
| 10 x 5 | 181 | 154 | 20363 | 91 | 117 | 91 |
| 5 x 10 | 130 | 102 | 9041 | 84 | 98 | 84 |
| 10 x 10 | 345 | 209 | 33466 | 108 | 136 | 108 |
| 15 x 10 | 527 | 338 | 209292 | 150 | 192 | 150 |
| 10 x 15 | 364 | 434 | 71882 | 123 | 135 | 123 |
| 17 x 13 | 779 | 591 | 975231 | 175 | 205 | 175 |

**(BLUE LINE - BFS, ORANGE LINE - DFS)**

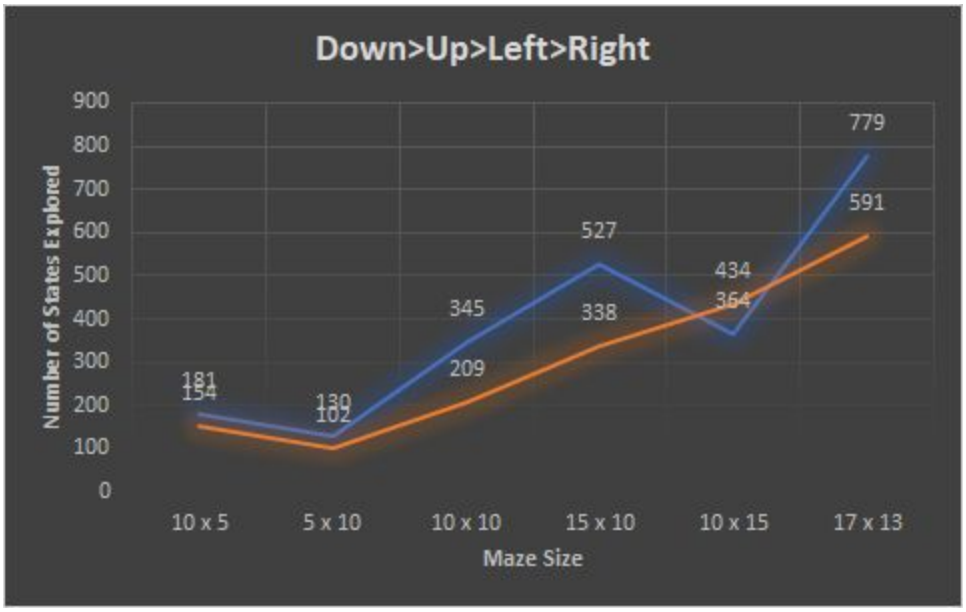


**Fig. Number of States explored vs. Maze Size**

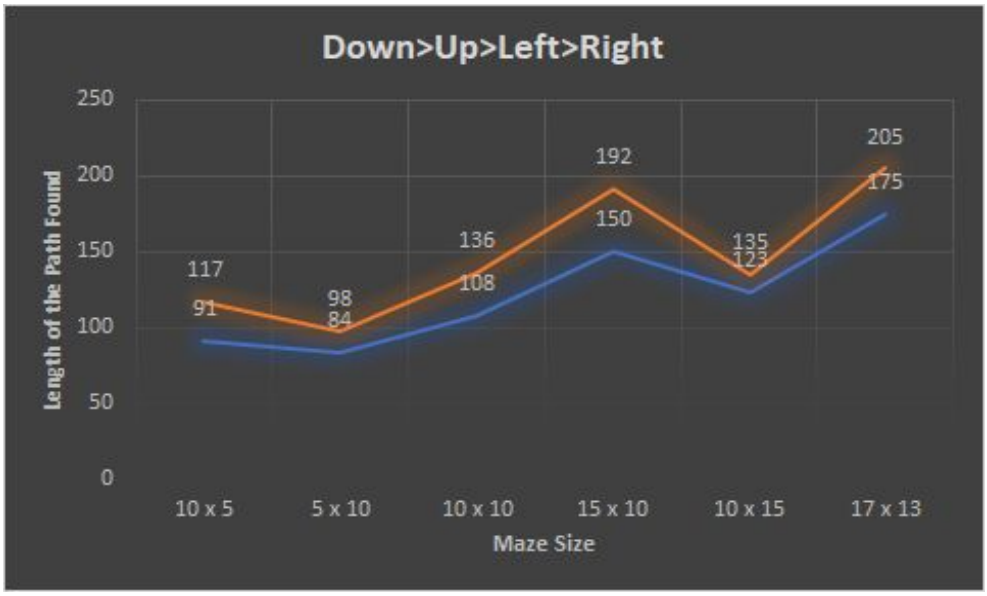


**Fig. Length of Path Found vs. Maze Size**

DFID

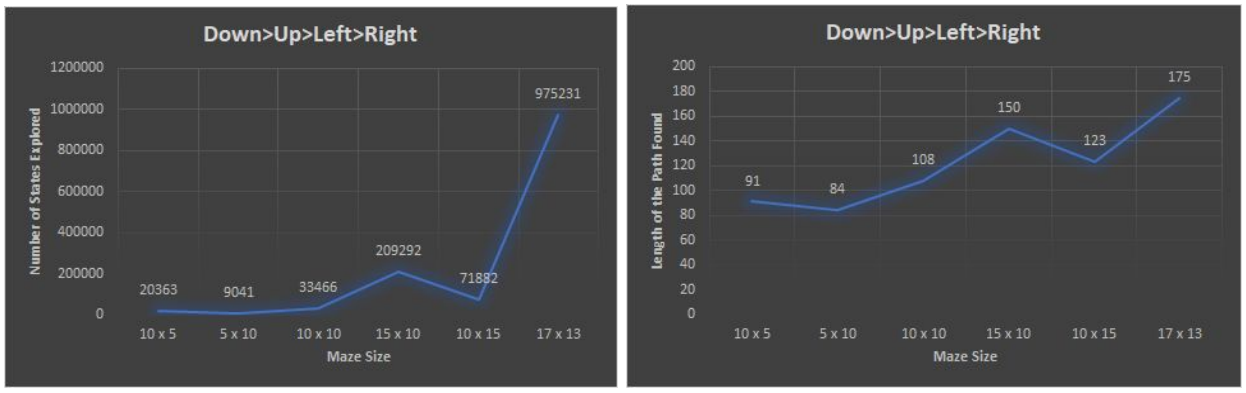


**Fig. Number of States explored vs. Maze Size (left)**
**Length of Path Found vs. Maze Size (right)**

## DEPENDENCE ON ORDER OF NEIGHBOURS

### Down > Up > Right > Left

| Maze Size | Number of States Explored | | | Length of the Path Found | | |
|---|---|---|---|---|---|---|
| Algorithm | BFS | DFS | DFID | BFS | DFS | DFID |
| 10 x 5 | 181 | 150 | 20383 | 91 | 117 | 91 |
| 5 x 10 | 130 | 100 | 9043 | 84 | 98 | 84 |
| 10 x 10 | 345 | 318 | 33285 | 108 | 136 | 108 |
| 15 x 10 | 528 | 324 | 209294 | 150 | 192 | 150 |
| 10 x 15 | 364 | 419 | 71891 | 123 | 135 | 123 |
| 17 x 13 | 779 | 697 | 975231 | 175 | 205 | 175 |

### Up > Down >Right > Left

| Maze Size | Number of States Explored | | | Length of the Path Found | | |
|---|---|---|---|---|---|---|
| Algorithm | BFS | DFS | DFID | BFS | DFS | DFID |
| 10 x 5 | 181 | 123 | 20545 | 91 | 117 | 91 |
| 5 x 10 | 130 | 102 | 9041 | 84 | 98 | 84 |
| 10 x 10 | 345 | 209 | 33466 | 108 | 136 | 108 |
| 15 x 10 | 527 | 338 | 162607 | 150 | 192 | 150 |
| 10 x 15 | 367 | 259 | 84579 | 123 | 135 | 123 |
| 17 x 13 | 779 | 568 | 1105223 | 175 | 205 | 175 |

### Right > Left > Up > Down

| Maze Size | Number of States Explored | | | Length of the Path Found | | |
|---|---|---|---|---|---|---|
| Algorithm | BFS | DFS | DFID | BFS | DFS | DFID |
| 10 x 5 | 181 | 131 | 42949 | 91 | 91 | 91 |
| 5 x 10 | 127 | 143 | 28813 | 84 | 84 | 84 |
| 10 x 10 | 346 | 114 | 90693 | 108 | 108 | 108 |
| 15 x 10 | 527 | 319 | 401883 | 150 | 150 | 150 |
| 10 x 15 | 363 | 378 | 95780 | 123 | 123 | 123 |
| 17 x 13 | 777 | 591 | 704391 | 175 | 205 | 175 |

## CONCLUSION

In the results section, it's observed that the trend of number of states explored is almost the same for BFS and DFID. This owes to the fact that both keep a notion of depth from the source node, which DFS does not follow and hence, has a different trend.

The dependence on order of neighbours added is summarised in the table below

| Algorithm | Dependence on the order of neighbours | |
|---|---|---|
| | Length of Path Found | No. of States explored |
| BFS | False | False |
| DFS | True | True |
| DFID | False | True |

The length of path found **does not** vary in BFS and DFID as they provide the optimal solution. Whereas, DFS returns a solution which is order dependent. This is expected as DFS does not take the depth from the source into consideration and blindly searches along a path.

The number of states explored **do not** vary (or very minimal) in BFS as it goes level by level, irrespective of the order in which the neighbours are added to the queue. Since DFS explores a different node, and in turn a different path every time the order is changed, its explored states also vary. As DFID is just an iterative DFS, even its explored states vary with the order of neighbours.