

# CS:312 Artificial Intelligence

## Assignment 1

### Group 7

Bhavesh Borse - 200010005, Eshita Pagare - 200010016

## 1 Introduction

The objective of this task is to simulate breadth-first search, depth-first search, and DFID in the state space. The state-space consists of an  $m \times n$  grid. The start state is (0,0). The goal state is the position of (\*) in the grid. The Pacman is allowed to move UP, DOWN, LEFT and RIGHT (except for boundary). Comparison of the number of states and path length between different search methods also between the orders in which neighbours are added.

## 2 Pseudo Codes

### 2.1 MoveGen()

```
def MoveGen(i, j, MAZE_input):
    global Open_Lists
    templist = []

    # This part is for the priority order of UP > DOWN > RIGHT > LEFT
    if(i < column-1):
        if ((MAZE_input[i+1][j] == ' ' or MAZE_input[i+1][j] == '*') and ((i+1,j) not in Open_Lists)):
            templist.append((i+1,j))
    if(i > 0):
        if (MAZE_input[i-1][j] == ' ' or MAZE_input[i-1][j] == '*') and ((i-1,j) not in Open_Lists):
            templist.append((i-1,j))
    if(j < column-1):
        if (MAZE_input[i][j+1] == ' ' or MAZE_input[i][j+1] == '*') and ((i,j+1) not in Open_Lists):
            templist.append((i,j+1))
    if(j > 0):
        if (MAZE_input[i][j-1] == ' ' or MAZE_input[i][j-1] == '*') and ((i,j-1) not in Open_Lists):
            templist.append((i,j-1))
    return templist
```

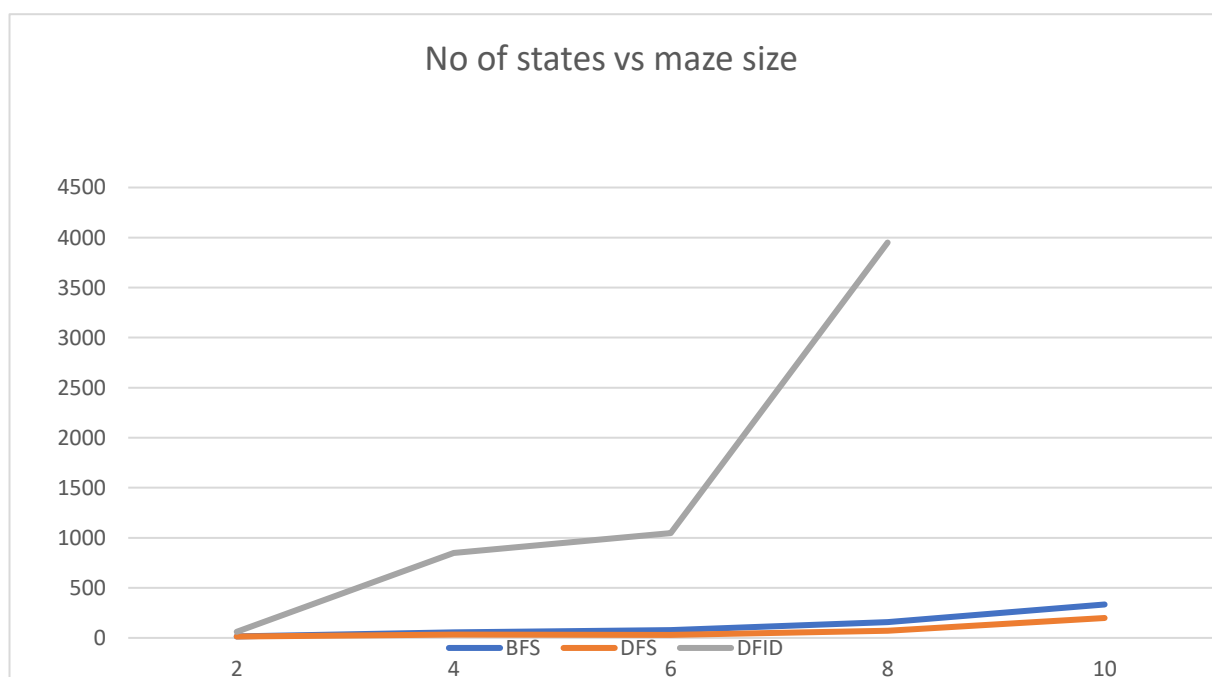
### 2.2 GoalTest()

```
def GoalTest(i, j, MAZE_input):
    # This function determines whether the coordinate (i,j) is the end goal for the pacman or not .
    if(MAZE_input[i][j] == '*'):
        return True
    else:
        return False
```

### 3 Maze results variation

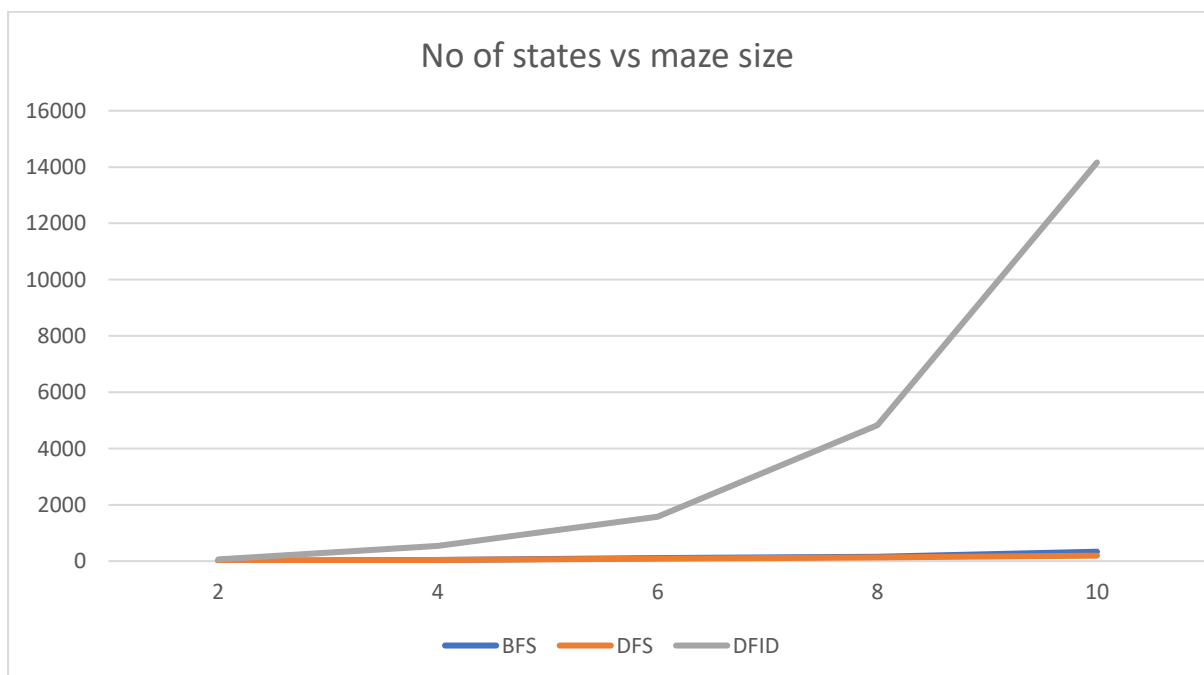
Following table shows statistics related to maze of different values of m and n and corresponding number of states and path length by using order: UP DOWN RIGHT LEFT  
Cell Width = 3 Cell height = 2

Algorithms	Order: UP > DOWN > RIGHT > LEFT			
	Horizontal cells	Vertical cells	No of states	Path length
BFS	2	2	15	10
DFS	2	2	14	10
DFID	2	2	61	10
BFS	4	4	54	32
DFS	4	4	32	32
DFID	4	4	849	32
BFS	6	6	78	30
DFS	6	6	30	30
DFID	6	6	1046	30
BFS	8	8	158	64
DFS	8	8	71	64
DFID	8	8	3950	64
BFS	10	10	334	90
DFS	10	10	199	90
DFID	10	10	11443	90



Following table shows statistics related to maze of different values of m and n and corresponding number of states and path length by using order: RIGHT LEFT DOWN UP  
Cell Width = 3 Cell height = 2

Algorithms	Order: RIGHT > LEFT > DOWN > UP			
	Horizontal cells	Vertical cells	No of states	Path length
BFS	2	2	13	10
DFS	2	2	12	10
DFID	2	2	62	10
BFS	4	4	42	24
DFS	4	4	43	28
DFID	4	4	544	28
BFS	6	6	107	34
DFS	6	6	88	40
DFID	6	6	1575	40
BFS	8	8	159	64
DFS	8	8	131	78
DFID	8	8	4832	78
BFS	10	10	335	90
DFS	10	10	194	108
DFID	10	10	14161	108



## 4 Dependence of Results:

The results of dependence of path length and number of states explored are summarized in the table below.

Algorithm	No of states	Path Length
BFS	True	False
DFS	True	True
DFID	True	True