

AI - LAB

Assignment-1

Report

Arvind Kumar M - 200020008

Tarun Saini - 200010051

December 26, 2021

Contents

1	Introduction	1
1.1	Variables used in program	1
1.2	Functions used in program	1
1.3	Class used in program	2
2	Pseudo Code for Movegen()	2
3	Pseudo Code for Goaltest()	2
4	Results	3
4.0.1	Output for sample test case : 1	3
4.0.2	Output for sample test case : 2	3
5	Comparision plot	4
6	Dependence of results on the order of neighbors added	5

1 Introduction

1.1 Variables used in program

- **choice** : variable to call respective function
- **Graph** [] : Stores the input maze in the form of array of strings
- **m** : Number of rows in input maze.
- **n** : Number of columns in input maze.
- **No of states** : Stores the number of states explored.
- **Length of path** : Stores the length of the path discovered.
- **closed** [] : Array to store the coordinate of states explored.
- **open_list** [] : Array to store neighbours which are discovered but not explored.
- **obj_list** [] : Array of classes for BFS.
- **obj_count** : Keeps track on number of objects added.
- **bfs_count** : Keeps track on number of objects explored.(The above three together works similar to queue.
- **current_position** : Stores the coordinates of current position in an array.
- **head** : class object to store the initial coordinate.

1.2 Functions used in program

- **movegen** () : Function that return the list of neighbours which are not explored and not in OPEN. Priority of neighbours added is DOWN >UP >RIGHT >LEFT
- **goaltest** () : Returns **True** if the packman has reached the food.
- **print_path** () : Adds 0s to the path found in the maze and prints the path to a output file.
- **dfs** () : Depth First Search implemented using CLOSED and OPEN list and class path.
- **bfs** () : Breadth First Search implemented using CLOSED, obj_list which works similar to queue using class path.
- **dfid** () : Implementing Depth First Iterative Deepening using dfs with incrementing depth in each loop. Function exits if the depth exceeds m*n.

1.3 Class used in program

class path: class members

- coordinate : To store the position [x,y]
- prev : To point parent node
- next : To point child node

2 Pseudo Code for Movegen()

```
def movegen(position):  
    list = []  
    if(DOWN != boundary and graph[DOWN] == ' ' or '*') and DOWN not  
        in OPEN):  
        list ← DOWN  
    if(UP != boundary and graph[UP] == ' ' or '*') and UP not in OPEN):  
        list ← UP  
    if(RIGHT != boundary and graph[RIGHT] == ' ' or '*') and RIGHT not in OPEN):  
        list ← RIGHT  
    if(LEFT != boundary and graph[LEFT] == ' ' or '*') and LEFT not in OPEN):  
        list ← LEFT  
    return list
```

3 Pseudo Code for Goaltest()

```
def goaltest(position):  
    x, y = position  
    if(GRAPH[x][y] == '*'):  
        return True  
    else:  
        return False
```

4 Results

4.0.1 Output for sample test case : 1

Input	Output - BFS	DFS	DFID
	42	24	442
	24	24	24
+--+--+--+--+	0--+--+--+--+	0--+--+--+--+	0--+--+--+--+
	00 0000	00 0000	00 0000
+ + + + +	+0 +0 +0 + +	+0 +0 +0 + +	+0 +0 +0 + +
	0000 0	0000 0	0000 0
+--+--+ +--+	+--+--+0 +--+	+--+--+0 +--+	+--+--+0 +--+
	0000	0000	0000
+ + +--+ +	+ + +--+0 +	+ + +--+0 +	+ + +--+0 +
*	000	000	000
+--+--+--+--+	+--+--+--+--+	+--+--+--+--+	+--+--+--+--+

4.0.2 Output for sample test case : 2

Input	Output - BFS
	59
	33
+--+--+--+--+	0--+--+--+--+
	00000
+--+ + + +	+--+0 + + + +
	000
+ +--+ +--+ +	+ 0+--+ +--+ +
	0
+ +--+--+--+ +	+ 0+--+--+--+ +
	000000 0000
+ +--+ + + +	+ +--+0 +0 +0 +
*	0000 000
+--+--+--+--+	+--+--+--+--+

DFS

```

41
33
0---+---+---+---+
00000 |   |   |
+---+0 + + + +
| 000 |   |   |
+ 0+---+ +---+ +
| 0|   |   |   |
+ 0+---+---+---+ +
| 000000 |0000 |
+ +---+0 +0 +0 +
|   |0000 |000
+---+---+---+---+
  
```

DFID

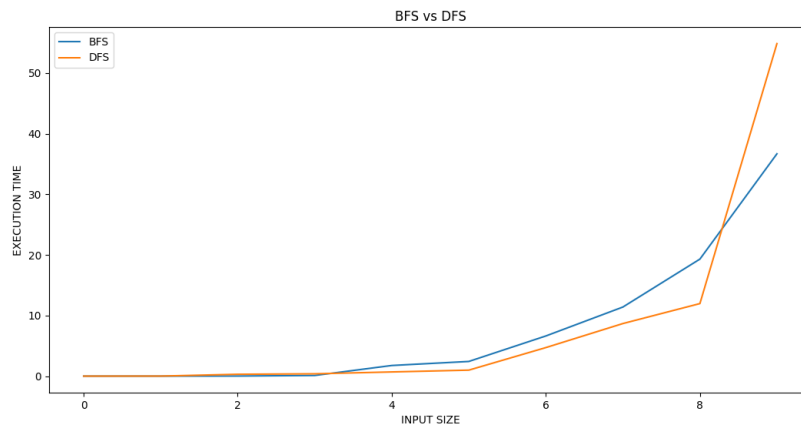
```

911
33
0---+---+---+---+
00000 |   |   |
+---+0 + + + +
| 000 |   |   |
+ 0+---+ +---+ +
| 0|   |   |   |
+ 0+---+---+---+ +
| 000000 |0000 |
+ +---+0 +0 +0 +
|   |0000 |000
+---+---+---+---+
  
```

5 Comparison plot

BFS vs DFS

When BFS and DFS are plotted for increasing input size



DFID

DFID takes more time compared to dfs and bfs but with less space complexity.

6 Dependence of results on the order of neighbors added

Input

```

+---+---+---+
|   |   |   |   |
+   +   +   +   +
|   |   |   |   |
+---+---+   +---+
|   |   |   |   |
+   +   +---+   +
|   |   |   |   *
+---+---+---+

```

BFS

- For BFS, number of states explored is only marginally dependent on the order of neighbours added.
- For sample case 1, number of states explored is either 42 or 43 for all preference orders.

42	D>U>R>L	
42	U>D>R>L	
43	D>U>L>R	D - DOWN
43	U>D>L>R	U - UP
42	L>R>U>D	R - RIGHT
42	R>L>U>D	L - LEFT
42	R>L>D>U	
42	L>R>D>U	

DFS

- For DFS, number of states explored is highly dependent on the order of neighbours added.
- For sample case 1, number of states explored is less when DOWN and RIGHT have higher preference
- And it is more when UP and LEFT have higher preference.

24	D>U>R>L	
24	U>D>R>L	
41	D>U>L>R	D - DOWN
41	U>D>L>R	U - UP
46	L>R>U>D	R - RIGHT
33	R>L>U>D	L - LEFT
33	R>L>D>U	
46	L>R>D>U	

DFID

- For DFID, dependancy of order of neighbours added is similar to DFS .

442	D>U>R>L	
442	U>D>R>L	
444	D>U>L>R	D - DOWN
444	U>D>L>R	U - UP
502	L>R>U>D	R - RIGHT
506	R>L>U>D	L - LEFT
506	R>L>D>U	
502	L>R>D>U	