CS 312: Artificial Intelligence Laboratory

Lab 1 Report

Balsher Singh – 180010008

Karan Sharma - 180010019

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 x 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). A comparison of the path length and the number of states explored between the different search methods and, also, between the orders in which neighbours are added, are performed.

Directions to Run Code →

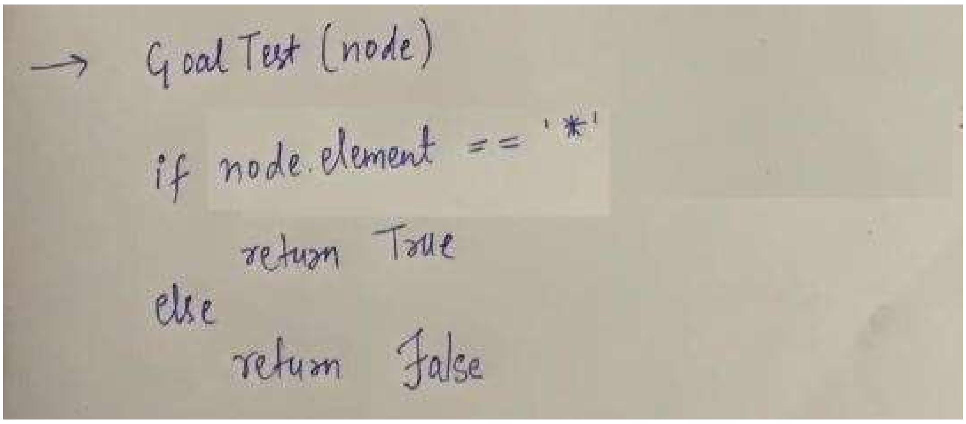
Command: **python Group4_Lab_1.py <input file>**

This will print output on terminal as well as in a file named 'output.txt'

2. Pseudo Code →

2.1 GoalTest(state)

Returns true if the input state is goal and false otherwise.



2.2 MoveGen(state)

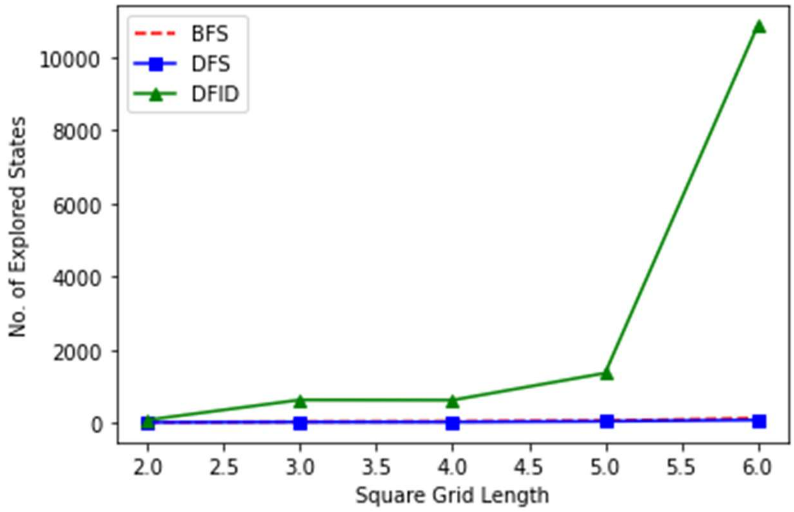
The function takes a state as input and returns a set of states that are reachable from the input state in one step or basically, it returns neighbours of the state.

```
→ MoveGen ( neighbour , node)
  # neighbour to store set of child
  # node is whose neighbour we have to find
  if down state is valid:
      neighbour. append ( below child)

  if above state is valid
      neighbour. append ( above child)

  if right state is valid
      neighbour. append ( right child)

  if left state is valid
      neighbour. append ( left child)

  return neighbour
```

3. Results, Statistics and Plots

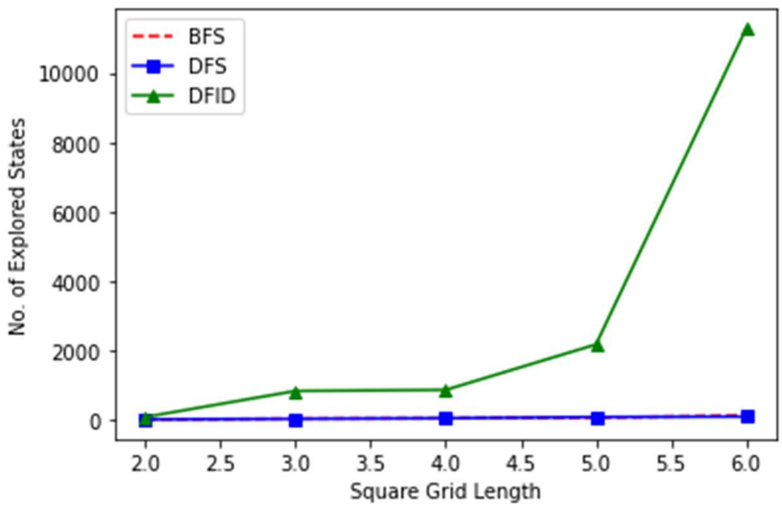
3.1 Order: Down, Up, Right, Left

Algorithm	*Statistics (cell width=3, cell height=2)*			
	No. horizontal cells	*No. vertical cells*	*No. states explored*	*Path length*
BFS	*2*	*2*	*15*	*10*
DFS	*2*	*2*	*14*	*10*
DFID	*2*	*2*	*80*	*10*
BFS	*3*	*3*	*35*	*23*
DFS	*3*	*3*	*23*	*23*
DFID	*3*	*3*	*627*	*23*
BFS	*4*	*4*	*42*	*24*
DFS	*4*	*4*	*24*	*24*
DFID	*4*	*4*	*621*	*24*
BFS	*5*	*5*	*59*	*33*
DFS	*5*	*5*	*41*	*33*
DFID	*5*	*5*	*1358*	*33*
BFS	*6*	*6*	*127*	*50*
DFS	*6*	*6*	*77*	*50*
DFID	*6*	*6*	*10846*	*50*



3.2 Order: Left, Right, Up, Down

Algorithm	*Statistics (cell width=3, cell height=2)*			
	No. horizontal cells	*No. vertical cells*	*No. states explored*	*Path length*
BFS	*2*	*2*	*13*	*10*
DFS	*2*	*2*	*11*	*10*
DFID	*2*	*2*	*81*	*10*
BFS	*3*	*3*	*35*	*23*
DFS	*3*	*3*	*29*	*29*
DFID	*3*	*3*	*832*	*23*
BFS	*4*	*4*	*42*	*24*
DFS	*4*	*4*	*46*	*26*
DFID	*4*	*4*	*862*	*24*
BFS	*5*	*5*	*59*	*33*
DFS	*5*	*5*	*82*	*37*
DFID	*5*	*5*	*2172*	*33*
BFS	*6*	*6*	*127*	*50*
DFS	*6*	*6*	*92*	*62*
DFID	*6*	*6*	*11274*	*50*



4. Conclusion

The results of the dependence of the path length and number of states explored, as seen in the previous section, are summarized in the table below. For small inputs in DFID, we observe that the increase in the number of explored states is due to the small branching factor and high constant attached with the time complexity.

Algorithm	***Dependence on order of neighbours added***	
	No. States Explored	***Path Length***
BFS	*True*	*False*
DFS	*True*	*True*
DFID	*True*	*False*