**CS 312: Artificial Intelligence Laboratory**

**Lab 1 Report**

**Online Group - 10**

**Rudraksh Sattabhayya – 190030038**

**Sumedhsing Rajput – 190030042**

**Introduction:**

The Aim of this Lab to achieve the 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:** 1: g++ Lab1\_OnlineGroup10.cpp -o showOutput.exe

2:showOutput.exe

This will print output on terminal

**1. Pseudo Code**

**1.1 GoalTest(state)**

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

**Algorithm**

1: **procedure** goalTest(state)

2: **if** state.value == “∗ “ **then**

3: **return** true

4: **return** false (=> state is not goal)

**1.2 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.

**Algorithm**

1: **procedure** moveGen(state)

2: *nextStates* ← () (=>initialize nextStates to empty set )

3: **for** neighbour n of *state* in order(DOWN,UP,RIGHT,LEFT) **do**

4: **if** n is not boundary **then**

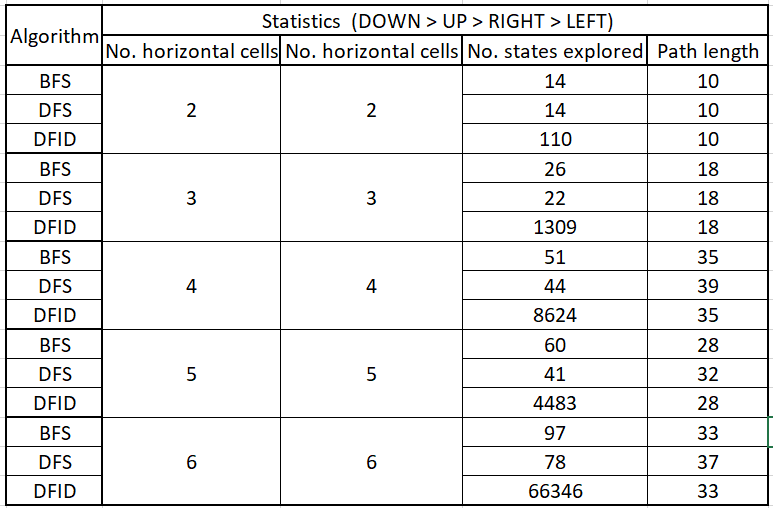
5: *nextStates.append(n)*

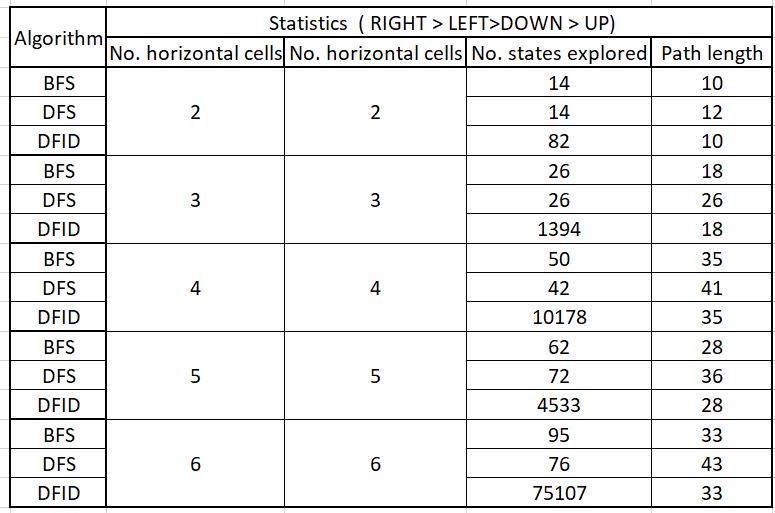
6: **return** *nextStates* (nextStates are required moves generated)

**2. Results**

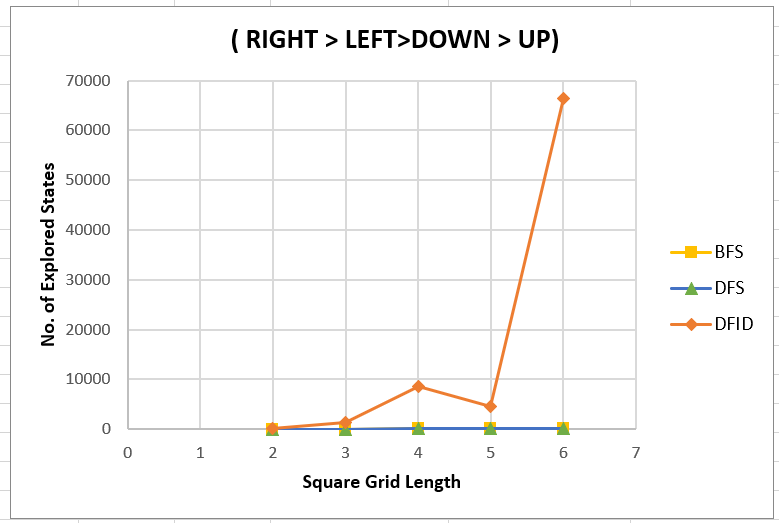
(\*All the mazes used for making this report are in mazes.txt)

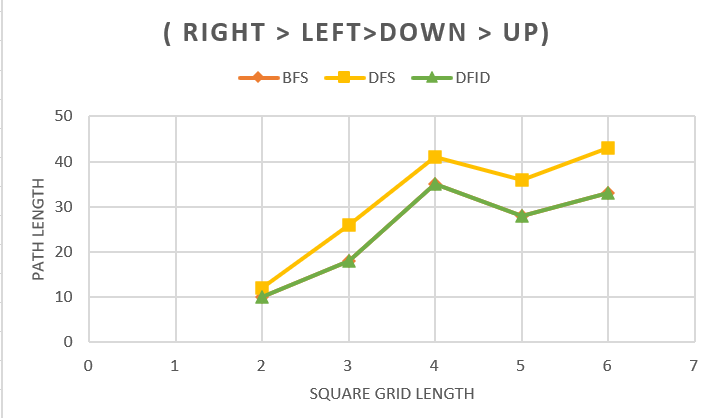
**2.1 Tables**





**2.2 Plots**

****

****

**3. 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.

