Name: Praveen Kumar K [Github](https://github.com/Pravi16/AI-LAB)

Date: 04/02/2022

Subject: AI

EXP3. BFS and DFS Algorithm

**AIM:**

In this experiment we will implement the Breadth First Search and depth First Search algorithms using python

**DESCRIPTION:**

**BFS:**

Breadth- rst search is a graph traversal algorithm that starts

traversing the graph from the root node and explores all the

neighbouring nodes. Then, it selects the nearest node and explores

all the unexplored nodes. While using BFS for traversal, any node

in the graph can be considered as the root node.

**DFS:**

Depth rst search (DFS) algorithm starts with the initial node of the

graph G, and then goes to deeper and deeper until we nd the goal

node or the node which has no children. The algorithm, then

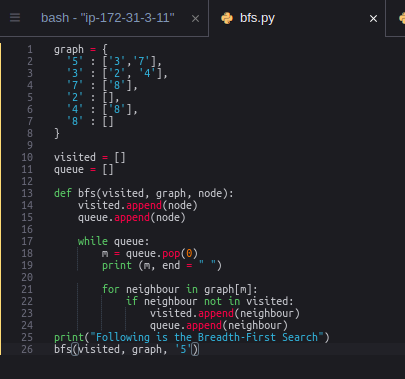
backtracks from the dead end towards the most recent node that

is yet to be completely unexplored.

The data structure which is being used in DFS is stack.

**Screenshots:**

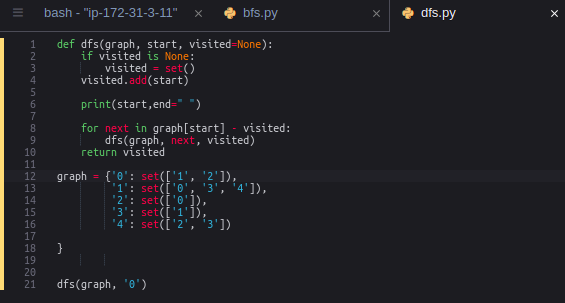
**BFS:**



**Output:**



**DFS:**



**Output:**



**Algorithm:**

**BFS**:

The steps involved in the BFS algorithm to explore a graph are

given as follows -

Step 1: SET STATUS = 1 (ready state) for each node in G

Step 2: Enqueue the starting node A and set its STATUS = 2

(waiting state)

Step 3: Repeat Steps 4 and 5 until QUEUE is empty

Step 4: Dequeue a node N. Process it and set its STATUS = 3

(processed state).

Step 5: Enqueue all the neighbours of N that are in the ready state

(whose STATUS = 1) and set

their STATUS = 2

(waiting state)[END OF LOOP]

Step 6: EXIT

**DFS:**

The steps involved in the DFS algorithm to explore a graph are

given as follows -

◦ Step 1: SET STATUS = 1 (ready state) for each node in G

◦ Step 2: Push the starting node A on the stack and set its

STATUS = 2 (waiting state)

◦ Step 3: Repeat Steps 4 and 5 until STACK is empty

◦ Step 4: Pop the top node N. Process it and set its STATUS =

3 (processed state)

◦ Step 5: Push on the stack all the neighbours of N that are in

the ready state (whose STATUS = 1) and set their

STATUS = 2 (waiting state)

[END OF LOOP]

◦ Step 6: EXIT

**Result:**

BFS and DFS are successfully implemented using python