

**CMR INSTITUTE OF TECHNOLOGY**

(UGC AUTONOMOUS)

Approved by AICTE | Accredited by NAAC with 'A' Grade
All B. Tech Programs Accredited by NBA

STUDENT REPORT

DETAILS

Name

DODDAPUDI SAI KUMAR

Roll Number

20R01A6614

EXPERIMENT

Title

WEEK 2

Description

EXPERIMENT NO: 2

Write a program to implement DFS Traversal.

ALGORITHM:

Depth First Search (DFS) algorithm traverses a graph in a depthward motion and uses a stack to remember to get the next vertex to start a search, when a dead end occurs in any iteration.

- **Rule 1** – Visit the adjacent unvisited vertex. Mark it as visited. Display it. Push it in a stack.
- **Rule 2** – If no adjacent vertex is found, pop up a vertex from the stack. (It will pop up all the vertices from the stack, which do not have adjacent vertices.)
- **Rule 3** – Repeat Rule 1 and Rule 2 until the stack is empty.

Source Code:

```
from collections import defaultdict
class Graph:
    def __init__(self):
        self.graph = defaultdict(list)
    def addEdge(self,u,v):
        self.graph[u].append(v)
    def DFSUtil(self, v, visited):
        visited.add(v)
        print(v, end=' ')
        for neighbour in self.graph[v]:
            if neighbour not in visited:
                self.DFSUtil(neighbour, visited)
    def DFS(self,v):
        visited=set()
        self.DFSUtil(v,visited)

g=Graph()
g.addEdge(0,1)
g.addEdge(0,2)
g.addEdge(1,2)
g.addEdge(2,0)
g.addEdge(2,3)
g.addEdge(3,3)
print("following is Depth First Traversal " "(starting from vertex 2)")
g.DFS(2)
```

RESULT

0 / 0 Test Cases Passed | 100

VIVA RESPONSES

Q) Depth First Search is equivalent to which of the traversal in the Binary Trees?

Pre-order Traversal

Q) Time Complexity of DFS is? (V – number of vertices, E – number of edges)

$O(V + E)$

Q) The Data structure used in standard implementation of Depth First Search is?

Stack

Q) When the Depth First Search of a graph is unique?

When the graph is a Linked List

Q) In Depth First Search, how many times a node is visited?

Equivalent to number of indegree of the node