Exp No. 2
Date:

Depth First Search.

Aim:

To implement Depth First Search (DFS) to traverse a graph.

Algorithm:

Step 1: Start

Step 2: Intialize an empty ~~set~~ stack and a list.

Step 3: Push the starting node onto stack.

Step 4: If stack != empty repeat setep 5 & 4.

Step 5: Pop the stack.

Step 6: Print the popped node.

Step 7: For each => for adjacent unvisited node.

Step 8: Make the neighbors visited.

Step 9: Push the unvisited node.

Step 10: Repeat until all nodes are visited.

Step 11: Stop.

Program:

```
def dfs(g, s):

    s = [start]
    V = ~~s()~~ set()
    while s:
        n = s.pop()

    if node not visited:
        print(node, end=" ")
        visited.add(node)
        for neighbour in ~~graph~~ g[node]
            if neighbour not in visited:
                s.append(neighbour)

graph = {
    'A' : ['B','C']
    'B' : ['D','E']
    'C' : ['F']
    'D' : []
    'E' : ['F']
    'F' : []
}

print("DFS traversal starting from node 'A'.")
dfs(g, 'A')
```

A