EX.NO:2.

DEPTH FIRST SEARCH

AIM:-

To implement depth first search (DFS) to traverse a graph and explore all vertices by visiting as far along each branch as possible before backtracking

ALGORITHM:-

Step 1: Start
Step 2: Initialize an empty stack and a list to keep track of visited nodes.
Step 3: Push the starting node onto stack & mark repeated.
Step 4: Pop the top node from the stack.
Step 5: Print or process the popped node.
Step 6: Mark the neighbors as visited.
Step 7: repeat until all reachable node are visited
Step 8: Stop.

PROGRAM:

```
def dfs(graph, start):
    stack = [start]
    visited = set().
    while stack:
        node = stack.pop()
        if node not in visited
            print(node, end = " ")
```

```
visited.add(node)
for neighbour in graph[node]:
    if neighbour not in visited:
        stack.append(neighbour)

graph = {
    'A': ['B', 'C'],
    'B': ['D', 'E'],
    'C': ['F'],
    'D': [],
    'E': ['F'],
    'F': []
}
print("DFS traversal starting from node 'A':")
dfs(graph, 'A')
```

OUTPUT:-

DFS traversal starting from node 'A' ACFBED

Result:-

The program is successfully executed and the output is verified.