Ex: 3
Date:

Depth First search

Aim:

To traverse a graph or tree starting from a node & visiting depth

Algorithm:

✱ Created a visited set
✱ created a stack & push the starting node
✱ mark the starting node
✱ when stack is not empty
    pop a node & print the node, & push
✱ terminate when stack is empty

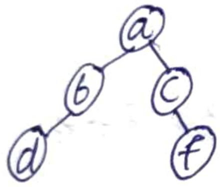
Code:

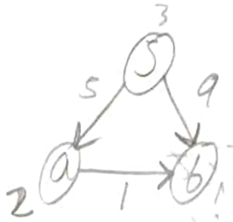
```
def dfs (graph, start)
    visited = set()
    stack = [start]
    visited.add(start)

    while stack:
        vertex = stack.pop()
        print(vertex, end=" ")

        for neighbor. in reversed (graph [vertex]):
            if neighbor not in visited:
                visited.add(neighbor)
                stack.append(neighbor)
graph = {}
n = int(input("enter no of nodes"))
```

```
for i in range(n):
        node = input(f"Enter node {i+1}:")
        neighbor = input(f" enter neighbor of {node}").split()
        graph[node] = neighbors
start_node = input("enter the starting node:")
dfs(graph, start_node)
```

Output:

For graph





```
Enter number of nodes: 5
enter node 1: a
Enter neighbors of a: b.c
enter node 2: b
Enter neighbors of b: d
Enter node 3: c
Enter neighbors of c: f
enter node 4: d
Enter neighbors of d:
enter node 5: f
enter neighbor to f:
Enter starting node: a
a b d c F
```

Result:

thus Depth first search algorith is successfully executed & output is verified.