

EX 4B  
25/02/21

## Depth First Search Traversal

**Aim:** To Study and Implement Depth First Search Traversal

### Algorithm:

The whole process can be categorized as follows:

1. Create a Class Graph which represents a Directed Graph Using Adjacency list representation.
2. Within this class initialize a constructor and declare a default dictionary to store the graph.
3. Create a Function named addEdge to add an edge to the graph.
4. Create another function which checks which is node is visited and not visited.
5. Mark the current Node as visited and Print it.
6. Perform the same recursively for all its adjacent Nodes.
7. Then create a Function DFS to do the actual traversal
8. Create a set to store visited vertices
9. Next, call the previously created Recursive Function, To print the Traversal
10. Create a graph based on the inputs given.
11. Create an input for the users, to input the from and to vertex in the graph.
12. Print the Traversal

### Program:

```
graph = {"A":["D","C","B"],
        "B":["E"],
        "C":["G","F"],
        "D":["H"],
        "E":["I"],
        "F":["J"]}
def dfs_non_recursive(graph, source):
    if source is None or source not in graph:
        return "Invalid input"
    path = []
    stack = [source]
    while(len(stack) != 0):

        print("\n',stack)
        s = stack.pop()
        if s not in path:
```

```

        print(s,end=' - ')
        path.append(s)
    if s not in graph:
        #leaf node
        continue
    for neighbor in graph[s]:
        print(neighbor,end=' ')
        stack.append(neighbor)

    print('\n')
    return " ".join(path)
DFS_path = dfs_non_recursive(graph, "A")
print("Depth First Path : ",end="")
print(DFS_path)

```

### Output:

```

['A']
A - D C B
['D', 'C', 'B']
B - E
['D', 'C', 'E']
E - I
['D', 'C', 'I']
I -
['D', 'C']
C - G F
['D', 'G', 'F']
F - J
['D', 'G', 'J']
J -
['D', 'G']
G -
['D']
D - H
['H']
H -

Depth First Path : A B E I C F J G D H

...Program finished with exit code 0
Press ENTER to exit console.

```

**Result:** We have successfully studied and implemented Depth First Search Traversal