

Roll. No. A016	Name: Varun Khadayate
Class B.Tech CsBs	Batch: 1
Date of Experiment: 3-1-2021	Date of Submission: 31-1-2022

To implement Depth First Search

Theory

The DFS algorithm is a recursive algorithm that uses the idea of backtracking. It involves exhaustive searches of all the nodes by going ahead, if possible, else by backtracking.

Here, the word backtrack means that when you are moving forward and there are no more nodes along the current path, you move backwards on the same path to find nodes to traverse. All the nodes will be visited on the current path till all the unvisited nodes have been traversed after which the next path will be selected.

This recursive nature of DFS can be implemented using stacks.

The basic idea is as follows:

- Pick a starting node and push all its adjacent nodes into a stack.
- Pop a node from stack to select the next node to visit and push all its adjacent nodes into a stack.
- Repeat this process until the stack is empty.

However, ensure that the nodes that are visited are marked. This will prevent you from visiting the same node more than once. If you do not mark the nodes that are visited and you visit the same node more than once, you may end up in an infinite loop.

Complexity

Time complexity $O(V+E)$, when implemented using an adjacency list.

Code

```
#include <stdio.h>
#include <stdlib.h>

int n; // number of nodes in graph
int *visited; // array to keep a track of nodes visited

void dfs(int i, int g[n][n]);

int main()
{
    int i, j, node;

    printf("Enter the number of nodes: ");
    scanf("%d", &n);
```

```

    int g[n][n];
    visited = (int *)calloc(n, sizeof(int));

    printf("Enter the adjacency matrix of the graph:\n");
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n; j++)
            scanf("%d", &g[i][j]);
    }

    printf("Enter the source node: ");
    scanf("%d", &node);
    printf("\nThe nodes reachable from %d are:\n", node);
    dfs(node-1, g);

    return 0;
}

void dfs(int i, int g[n][n])
{
    int j;

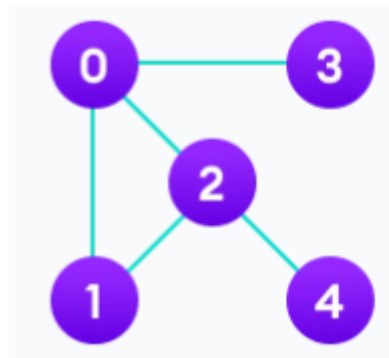
    visited[i] = 1;
    printf("%d ", i+1);

    for (j = 0; j < n; j++)
    {
        if (!visited[j] && g[i][j] == 1)
            dfs(j,g);
    }
}

```

Output

Taking this example



```
"E:\TY\SEM VI\AI\DFS.exe"
Enter the number of nodes: 5
Enter the adjacency matrix of the graph:
0 1 1 1 0
1 0 1 0 0
1 1 0 0 1
1 0 0 0 0
0 0 1 0 0
Enter the source node: 1

The nodes reachable from 1 are:
1 2 3 5 4
Process returned 0 (0x0)   execution time : 75.766 s
Press any key to continue.
```