



আন্তর্জাতিক ইসলামী বিশ্ববিদ্যালয় চট্টগ্রাম
الجامعة الإسلامية العالمية شيتاغونغ
International Islamic University Chittagong

Lab Report

Course Code: CSE-2422

Course Title: Computer Algorithm Lab

Submitted By:

Name: Adrita Nandi

Id: C241474.

Semester: 4th.

section: 4CF

Department: Computer Science & Engineering

Submission date: 06/12/2025

Assignment: Lab 14 [Depth First Search]

Submitted To:

Miskatul Jannat Tuly
Lecturer, Dept. of CSE, IIUC.

Lab no:14

Lab Title: Depth First Search

Introduction:

Depth-First Search (DFS) is a graph traversal algorithm that explores as far as possible along a path before backtracking. It uses a stack (or recursion) and helps in solving problems like pathfinding, cycle detection, and component finding.

Objective:

- ☐ To understand the concept of Depth-First Search.
- ☐ To learn how DFS explores nodes deep first.
- ☐ To implement DFS using recursion.
- ☐ To observe how DFS traversal works on a graph.
- ☐ To analyze DFS output and behavior.

Code: https://github.com/ADRITA-NANDI/Algorithm_CSE-2422.git

```
*C241474_BFS.cpp x C241474_DFS.cpp x
#include <iostream>
#include <vector>
using namespace std;

void dfs(int node, vector<vector<int>>& graph, vector<bool>& visited) {
    visited[node] = true;
    cout << node << " ";

    for (int next : graph[node]) {
        if (!visited[next]) {
            dfs(next, graph, visited);
        }
    }
}

int main() {
    int n, e;
    cout << "Enter number of nodes: ";
    cin >> n;

    cout << "Enter number of edges: ";
    cin >> e;

    vector<vector<int>> graph(n);

    cout << "Enter edges (u v) for undirected graph:\n";
    for (int i = 0; i < e; i++) {
        int u, v;
        cin >> u >> v;
        graph[u].push_back(v);
        graph[v].push_back(u);
    }

    int start;
    cout << "Enter starting node: ";
    cin >> start;

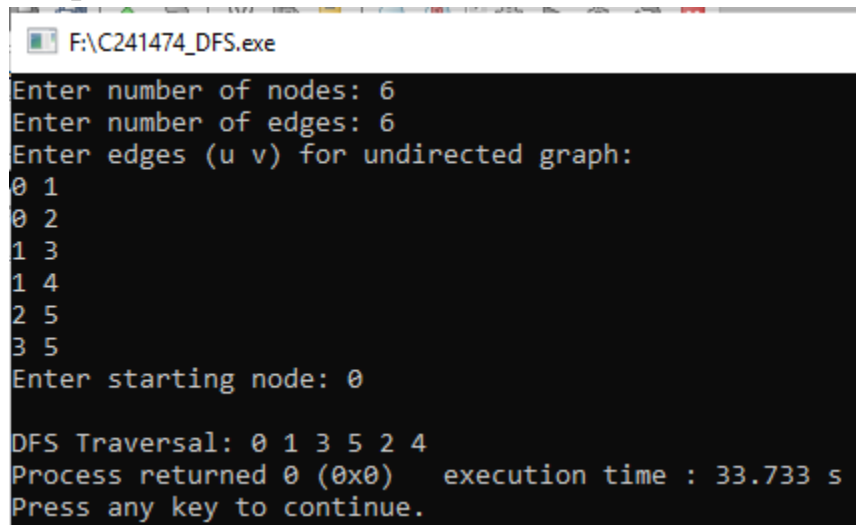
    int start;
    cout << "Enter starting node: ";
    cin >> start;

    vector<bool> visited(n, false);

    cout << "\nDFS Traversal: ";
    dfs(start, graph, visited);

    return 0;
}
```

Output:



```
F:\C241474_DFS.exe
Enter number of nodes: 6
Enter number of edges: 6
Enter edges (u v) for undirected graph:
0 1
0 2
1 3
1 4
2 5
3 5
Enter starting node: 0

DFS Traversal: 0 1 3 5 2 4
Process returned 0 (0x0)   execution time : 33.733 s
Press any key to continue.
```

Discussion:

The DFS program explored nodes by going deep into each path before backtracking. The output showed that DFS does not follow level order but follows depth-first order. This experiment helped understand how recursion works in DFS and why it is useful for exploring graph paths and detecting cycles.