Data Structure And Algorithum

Lab Report

Name:	Ch Hasnain Zafar
Registration #:	SEU-F16-104
Lab Report #:	09
Dated:	21-04-2018
Submitted To:	Mr. Usman Ahmed

The University of Lahore, Islamabad Campus
Department of Computer Science & Information Technology

Experiment # 1
Depth First Search Implementation

Objective
To understand and implement the DFS Problem.

Software Tool
1. DEV C++

1 Theory

Depth First Traversal (or Search) for a graph is similar to Depth First Traversal of a tree. The only catch here is, unlike trees, graphs may contain cycles, so we may come to the same node again. To avoid processing a node more than once, we use a boolean visited array.

2 Task

2.1 Procedure: Task 1

```
#include<iostream>
#include<conio.h>
#include<stdlib.h>
using namespace std;
int cost[10][10],i,j,k,n,stk[10],top,v,visit[10],visited[10];

main()
{
int m;
cout <<"enterno of vertices";
cin >> n;
cout <<"ente no of edges";
cin >> m;
cout <<"\nEDGES␣\n";
for(k=1;k<=m;k++)
{
cin >>i>>j;
cost[i][j]=1;
}

cout <<"enter␣initial␣vertex";
cin >>v;
cout <<"ORDER␣OF␣VISITED␣VERTICES";
cout << v <<"␣";
visited[v]=1;
k=1;
while(k<n)
{
for(j=n;j>=1;j--)
if(cost[v][j]!=0 && visited[j]!=1 && visit[j]!=1)
{
visit[j]=1;
stk[top]=j;
top++;
}
v=stk[--top];
cout<<v << "␣";
k++;
visit[v]=0; visited[v]=1;
}
}
```