A picture containing text

Description automatically generated



Lab Journal: 11

Date: 08 / 12 / 24

Student : Abdul Rafay

Enrollment : 01-131232-004

Department of Software Engineering

Bahria University, Islamabad

Data Structures & Algorithms Lab (Spring-2024)

Teacher: RAHEELA AMBRIN

**Comments:**

**Signature**

### Code:

All the code files are uploaded on GitHub: <https://github.com/CharlieFour/DSA_Lab>

You can check out the code on GitHub in Lab\_11 folder.

#### Graph.h

#pragma once   
  
#include <iostream>  
  
#define MAXVERTEXS 5  
  
struct edge  
{  
    bool adj;  
    int weight;  
  
    friend std::ostream& operator<<(std::ostream& os, const edge& edge)  
    {  
        os << "[" << edge.weight << "]";  
        return os;  
    }  
};  
  
class Graph  
{  
    private:  
        edge adjMatrix[MAXVERTEXS][MAXVERTEXS];  
    public:  
        Graph();  
  
        void addEdge(int vertex1, int vertex2, int weight);  
        void removeEdge(int vertex1, int vertex2);  
        void display();  
};

#### Graph.cpp

#include "graph.h"  
  
Graph::Graph()  
{  
    for (int i = 0; i < MAXVERTEXS; i++)  
    {  
        for (int j = 0; j < MAXVERTEXS; j++)  
        {  
            adjMatrix[i][j] = {0, 0};  
        }  
    }  
}  
void Graph::addEdge(int vertex1, int vertex2, int weight)  
{  
    adjMatrix[vertex1][vertex2].adj = true;  
    adjMatrix[vertex2][vertex1].adj = true;  
    adjMatrix[vertex1][vertex2].weight = weight;  
    adjMatrix[vertex2][vertex1].weight = weight;  
}  
void Graph::removeEdge(int vertex1, int vertex2)  
{  
    adjMatrix[vertex1][vertex2].adj = false;  
    adjMatrix[vertex2][vertex1].adj = false;  
    adjMatrix[vertex1][vertex2].weight = 0;  
    adjMatrix[vertex2][vertex1].weight = 0;  
}  
  
void Graph::display()  
{  
    for (int i = 0; i < MAXVERTEXS; i++)  
    {  
        for (int j = 0; j < MAXVERTEXS; j++)  
        {  
            std::cout << adjMatrix[i][j] << " ";  
        }  
        std::cout << std::endl;  
    }  
}

#### Main

#include <iostream>  
#include "..\lib\graph.h"  
  
using namespace std;  
  
int main()  
{  
    Graph graph;  
  
    graph.addEdge(0, 1, 5);  
    graph.addEdge(0, 2, 9);  
    graph.addEdge(1, 2, 4);  
    graph.addEdge(2, 3, 7);  
  
    graph.display();  
  
    graph.removeEdge(0, 2);  
    graph.removeEdge(2, 3);  
    cout << "\nAfter removing edges\n";  
    graph.display();  
  
    system("pause");  
    return 0;  
}

### Screen Shots:

A screenshot of a computer

Description automatically generated