 实 验 报 告

课程名称： 数据结构

题 目： 邻接矩阵的实现

专业班级： 18空间信息与数字技术

姓 名： 林培赟

学 号： 19号

指导老师： 袁莹

实验时间： 2019年12月11日

第三次实验报告

班级：18空间 姓名：林培赟 学号：1812001119

1. 实验目的及原理
2. 掌握图的逻辑结构；
3. 掌握图的邻接矩阵存储结构；
4. 验证图的邻接矩阵存储及其遍历操作的实现

二．实验内容

#pragma once

#ifndef MGraph\_H

#define MGraph\_H

const int MaxSize = 100;

template<class DataType>

class MGraph

{

public:

    MGraph(DataType a[], int n, int e);

    ~MGraph() {}

    void DFSTraverse(int v);

    void BFSTraverse(int v);

private:

    DataType vertex[MaxSize];

    int arc[MaxSize][MaxSize];

    int vertexNum, arcNum;

    int visited[MaxSize];

};

#endif

#include <iostream>

using namespace std;

#include"MGraph.h"

template<class DataType>

MGraph<DataType>::MGraph(DataType a[], int n, int e)

{

    int i, j, k;

    vertexNum = n, arcNum = e;

    for (i = 0; i < vertexNum; i++)

        vertex[i] = a[i];

    for (i = 0; i < vertexNum; i++)

        for (j = 0; j < vertexNum; j++)

            arc[i][j] = 0;

    for (k = 0; k < arcNum; k++)

    {

        cout << "请输入边的两个顶点的序号：";

        cin >> i >> j;

        arc[i][j] = 1; arc[j][i] = 1;

    }

}

template<class DataType>

void MGraph<DataType>::DFSTraverse(int v)

{

    cout << vertex[v];

    visited[v] = 1;

    for (int j = 0; j < vertexNum; j++)

        if (arc[v][j] == 1 && visited[j] == 0)

            DFSTraverse(j);

}

template<class DataType>

void MGraph<DataType>::BFSTraverse(int v)

{

    int Q[MaxSize];

    int w, j;

    int front = -1, rear = -1;

    cout << vertex[v];

    visited[v] = 1; Q[++rear] = v;

    while (front != rear)

    {

        w = Q[++front];

        for(j=0;j<vertexNum;j++)

            if (arc[w][j] == 1 && visited[j] == 0)

            {

                cout << vertex[j];

                visited[j] = 1;

                Q[++rear]=j;

            }

    }

}

#include<iostream>

using namespace std;

#include"MGraph.cpp"

int visited[MaxSize] = { 0 };

int main()

{

    char ch[6] = { 'A','B','C','D','E'};

    int i;

    MGraph<char>MG(ch, 5, 6);

    for (i = 0; i < MaxSize; i++)

        visited[i] = 0;

    cout << "深度优先遍历序列是：";

    MG.DFSTraverse(0);

    cout << endl;

    for (i = 0; i < MaxSize; i++)

        visited[i] = 0;

    cout << "广度优先遍历序列是：";

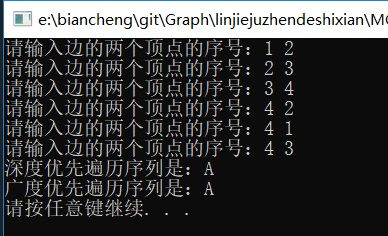
    MG.BFSTraverse(0);

    cout << endl;

    return 0;

}

1. 实验成果



1. 实验心得

本次实验中我完成了邻接矩阵的实现，掌握图的逻辑结构，掌握了图的邻接矩阵存储结构，验证了图的邻接矩阵存储及其遍历操作的实现。