南京信息工程大学 数据结构 实验(实习)报告

实验(实习)名称 **图的建立与遍历** 实验(实习)日期 2021.11.25 得分 指导老师 马瑞

系 **计软院** 专业  **计算机科学与技术** 班级 20（1） 姓名 鲁哲豪 学号 202083290400

一、实验目的



二、 实验内容与步骤







1.建立图G

#include<stdio.h>

#include<stdlib.h>

#define MaxM 1000001

#define MaxN 100001

typedef struct tEdge{

    int to,weight;

    struct tEdge\* next;

}Edge;

typedef struct tVertex{

    char data;

    Edge\* edges;

}Vertex;

Vertex v[MaxN];

int n,m;

void setEdge(a,b,w){

    Edge\* p=v[a].edges;

    while(p->next)

        p=p->next;

    p->to=b;

    p=p->next=malloc(sizeof(Edge));

    p->next=NULL;

}

int main(){

    scanf("%d%d",&n,&m);

    char s[100];

    scanf("%s",s+1);

    for(int i=1;i<=n;i++){

        v[i].data=s[i];

        v[i].edges=malloc(sizeof(Edge));

        v[i].edges->next=NULL;

    }

    for(int i=1;i<=m;i++){

        int a,b,w;

        scanf("%d%d%d",&a,&b,&w);

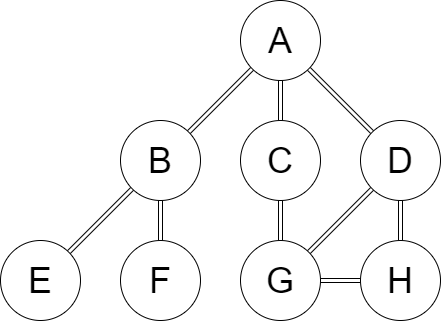
        setEdge(a,b,w);

        setEdge(b,a,w);

    }

    return 0;

}

测试数据：对应图G：

8 9

ABCDEFGH

1 2 1

1 3 1

1 4 1

2 5 1

2 6 1

3 7 1

4 7 1

4 8 1

7 8 1

2.dfs

使用递归算法，在邻接表的情况下是O(n+e)的（n为节点数，e为边数）

void dfs(int n){

    if(vis1[n])

        return;

    vis1[n]=1;

    printf("%c",v[n].data);

    Edge\* p=v[n].edges;

    while(p->next){

        dfs(p->to);

        p=p->next;

    }

}



结果符合预期

3.bfs

使用队列完成操作。复杂度也是O(n+e)的（n为节点数，e为边数）

    int queue[MaxN]={1},l=0,r=1;

    vis2[1]=1;

    for(;l<r;l++){

        printf("%c",v[queue[l]].data);

        Edge\* p=v[queue[l]].edges;

        for(;p->next;p=p->next){

            if(vis2[p->to])

                continue;

            vis2[p->to]=1;

            queue[r++]=p->to;

        }

    }



结果符合预期

三、 实验心得

**图**是由若干给定的顶点及连接两顶点的边所构成的图形，这种图形通常用来描述某些事物之间的某种特定关系。顶点用于代表事物，连接两顶点的边则用于表示两个事物间具有这种关系。

图的存储，为了节约空间，可用邻接表。

遍历是基本操作，可分深度优先/广度优先两种方法。

编写图论算法时，要用到前面学的队列之类的简单数据结构。