

实验5 图的物理实现

软件1801 肖云杰 201826010113

11月20日

- 根据书上的代码，开始代码的编写
- 成功完成书上Graph类和邻接矩阵表示法Graphm类的基本编写
- 注意的是，在assert中，我们做了相应修改适应代码
- 在UNVISITED中，我们默认让其等于0，使其满足其int的类型

11月21日

- 进行代码的初步运行调试
- 调试无误后，开始用我们的ADT去做该题
- 首先是数据的输入，考虑数据输入度数相同时的情况，做部分初定义

```
int n,m;
cin >> n>>m;
char name[n];
int chu[n];
for(int i=0;i<n;i++)
{
    cin >> name[i];
    chu[i]=0;
}
Graphm a(n);
for(int i=0;i<m;i++)
{
    char aa,bb;
    int a1=0,b1=0,c1=0,j=0;
    cin >> aa>>bb>>c1;
    for(j=0;j<n;j++)
    {
        if(aa==name[j]) a1=j;
        if(bb==name[j]) b1=j;
    }
    a.setEdge(a1,b1,c1);
}
```

- 数据输入后，简单输出邻接矩阵，判断，和自己想的没有问题，证明ADT实现没有问题

```

for(int i=0;i<n;i++)
{
    for(int j=0;j<n;j++)
    {
        cout << a.matrix[i][j]<<" ";
        if(a.matrix[i][j]!=0) chu[i]++;
    }
    cout << endl;
}

```

- 下面是数据的检验，考虑到度数相同的比较，自己做一个简单的字典排序（数字和char数对应）即可

```

char Ma='#';
int max=-1;
for(int i=0;i<n;i++)
{
    if(max<chu[i]){
        Ma=name[i];
        max=chu[i];
    }
    if(max==chu[i])
        Ma=(Ma>name[i])?Ma:name[i];
}
cout << Ma<<' '<<max;

```

一遍AC，没有什么问题

邻接表

11-21日

- 做完邻接矩阵后，一鼓作气，进军邻接表
- 考虑到邻接表的链表问题，采用的是STL模板容器的list容器，进行简单的判断即可
- 书上的函数是不知道哪里的List，不妨自己写
- 基本函数如下

```

class Graph1 : public Graph{
private:
    int numVert;
    std::list<int> ee[50];
public :
    Graph1(int numvert){
        this->numVert=numvert;
        for(int i=0;i<50;i++) ee[i].push_back(i);
    }
    ~Graph1(){
    }
    void build(int v1,int v2){

```

```

        std::list<int>::iterator it = find( ee[v1].begin(),
ee[v1].end(), v2 );
        if ( it == ee[v1].end() )
            ee[v1].push_back(v2);
    }
    void dele(int v1,int v2)
    {
        ee[v1].remove(v2);
    }
    void print(int n)                //检验测试邻接表
    {
        std::list<int>::iterator it = ee[n].begin();
        if(it==ee[n].end()){
            std::cout<<std::endl;
            return;
        }
        std::cout << *it;
        for(it++;it!=ee[n].end();it++)
        {
            std::cout << "->"<<*it;
        }
        std::cout<<std::endl;
    }
}

```

- 构造，析构，建表，删表，测试函数
- 自行测试无问题，一遍AC