项目说明文档

数据结构课程设计

——电网建设造价模拟系统

作 者 姓 名： 邓泉

学 号： 1953871

指 导 教 师： 张颖

学院、 专业： 软件学院 软件工程

同济大学

Tongji University

目 录

[1分析](#_Toc3970)

[1.1 题目背景](#_Toc13213)

[1.2背景分析](#_Toc21315)

[1.3 功能分析](#_Toc25756)

[2 设计](#_Toc25380)

[2.1 数据结构设计](#_Toc1413)

[2.2 成员与操作设计](#_Toc25443)

[2.3 系统设计](#_Toc24059)

[3.功能实现](#_Toc18443)

[3.1总流程实现](#_Toc30981)

[3.2 建立最小生成树](#_Toc25226)

3.3 其他成员函数的实现............................................................................................

[4 测试](#_Toc4819)

[4.1 功能测试](#_Toc21728)

[4.1.1 正常测试](#_Toc2478)

[4.2 边界测试](#_Toc24407)

[4.2.1 空图的最小生成树](#_Toc4949)

[4.3 出错测试](#_Toc25825)

[4.3.1 未生成最小生成树就要求显示](#_Toc16982)

[4.3.2 未知指令](#_Toc9111)

# 

# 1分析

## 题目背景

假设一个城市有n个小区，要实现n个小区之间的电网都能够相互接通，构造这个城市n个小区之间的电网，使总工程造价最低。请设计一个能够满足要求的造价方案。

## 1.2背景分析

如何在已有顶点之间建立一条通路，使得该通路连通所有顶点且长度最短，是一个在现实生活中非常常见的问题。解决这一问题有广泛而深远的现实意义。

## 1.3 功能分析

本程序要求可以在与用户的交互之中获得顶点和边的信息，并在需要时产生、显示已知图的最小生成树。

# 2 设计

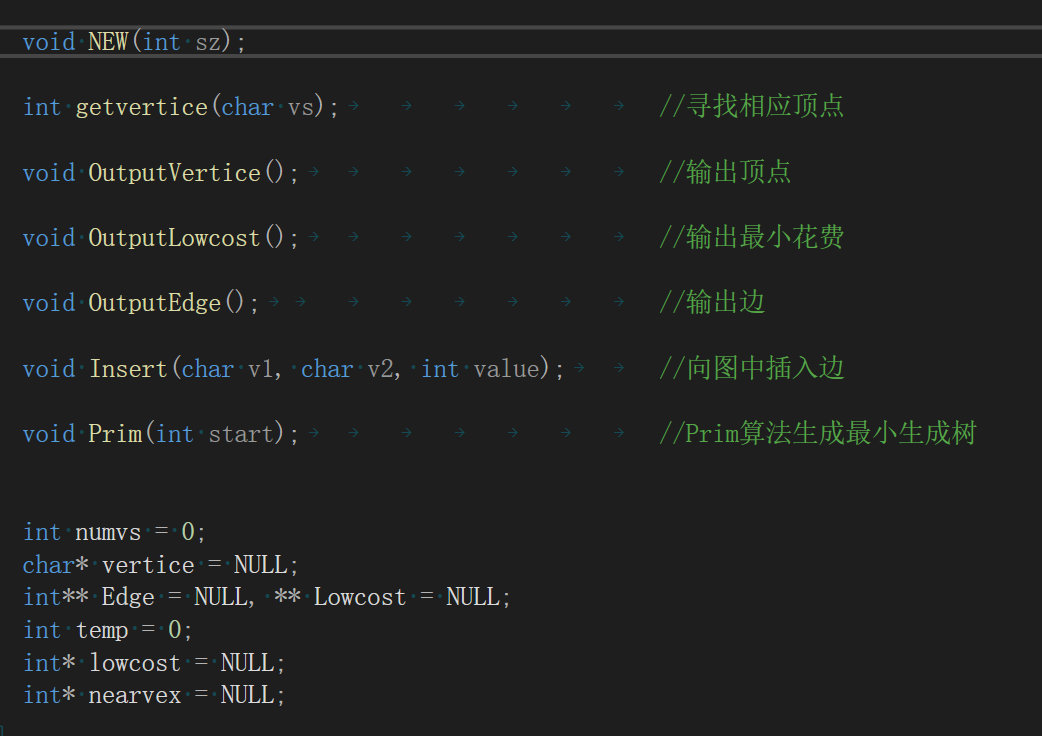
## 2.1 数据结构设计

现实生活中的城市、道路关系可以看作一张无向图，该问题可以抽象为求已知图的最小生成树问题。

用邻接矩阵来保存这张电网图，并附带辅助存储数组Lowcost和nearvex分别存放当前点到连通图的最小花费和其最小花费所到的点。使用Prim算法产生最小生成树。

## 2.2 成员与操作设计

**EWNet类**

****

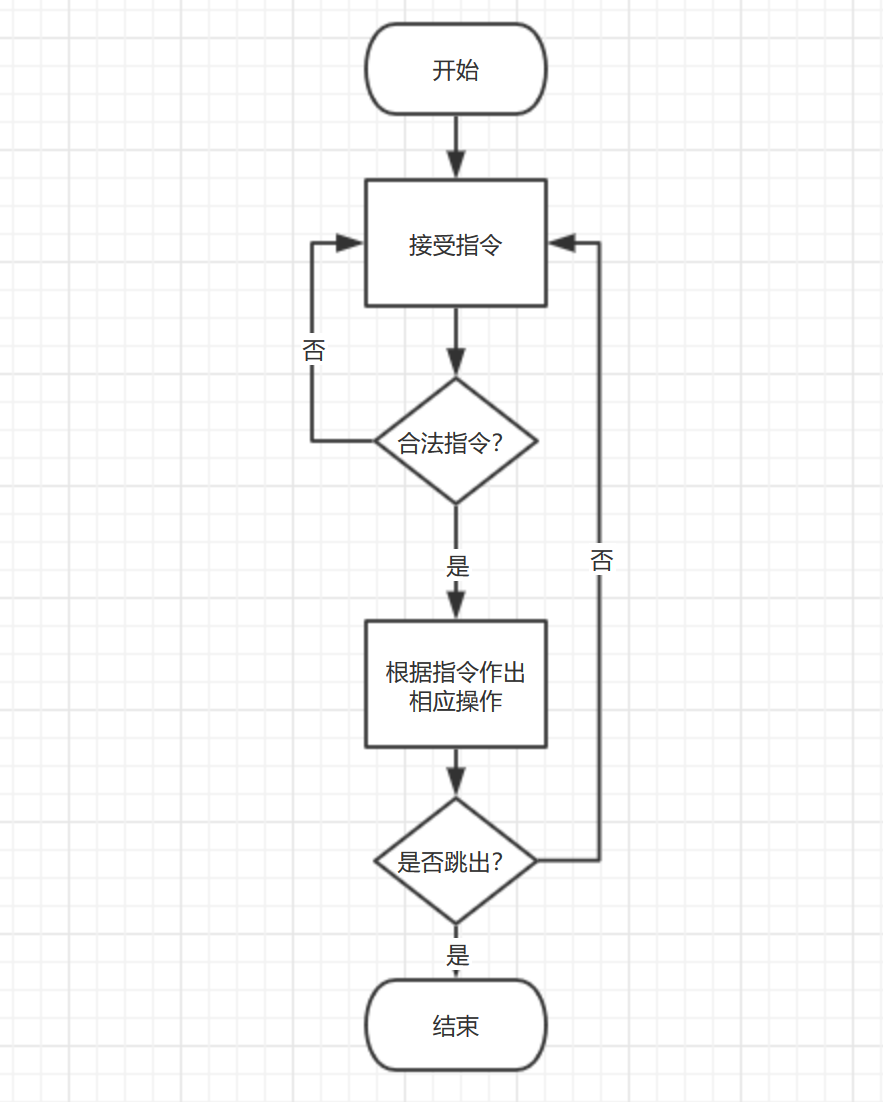
## 2.3 系统设计

系统通过交互式操作，不断根据用户要求对程序内保存的图进行修改，并在需要时建立、显示最小生成树。

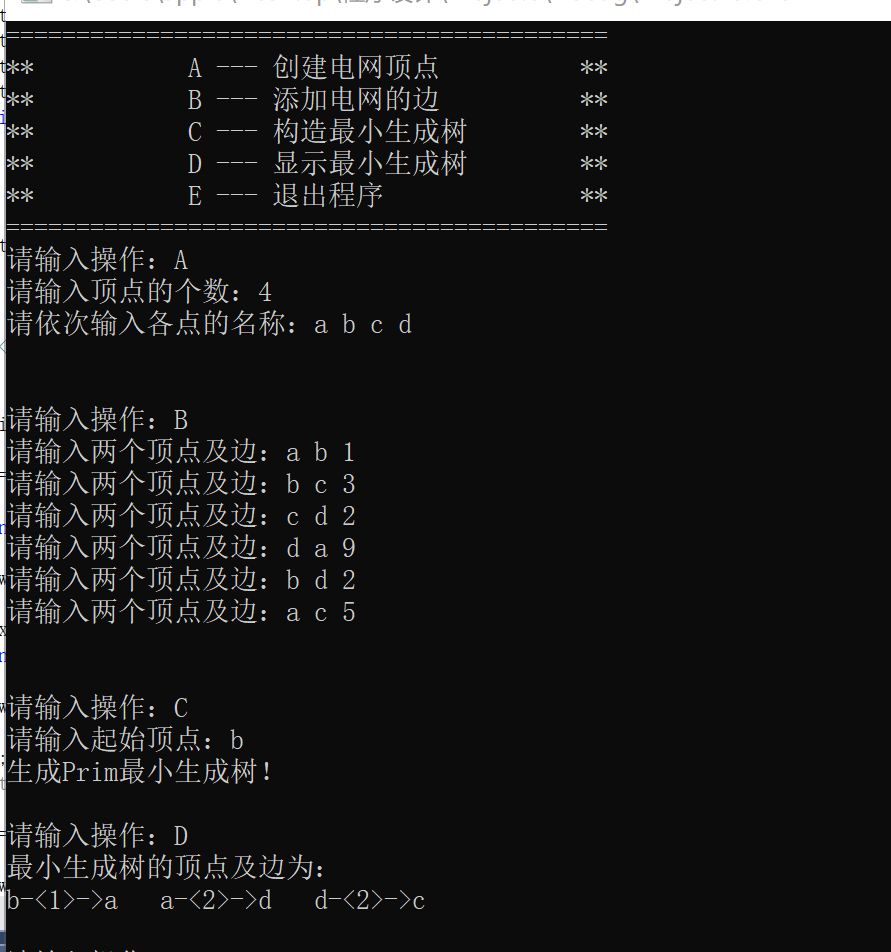
# 3.功能实现

## 3.1总流程实现：

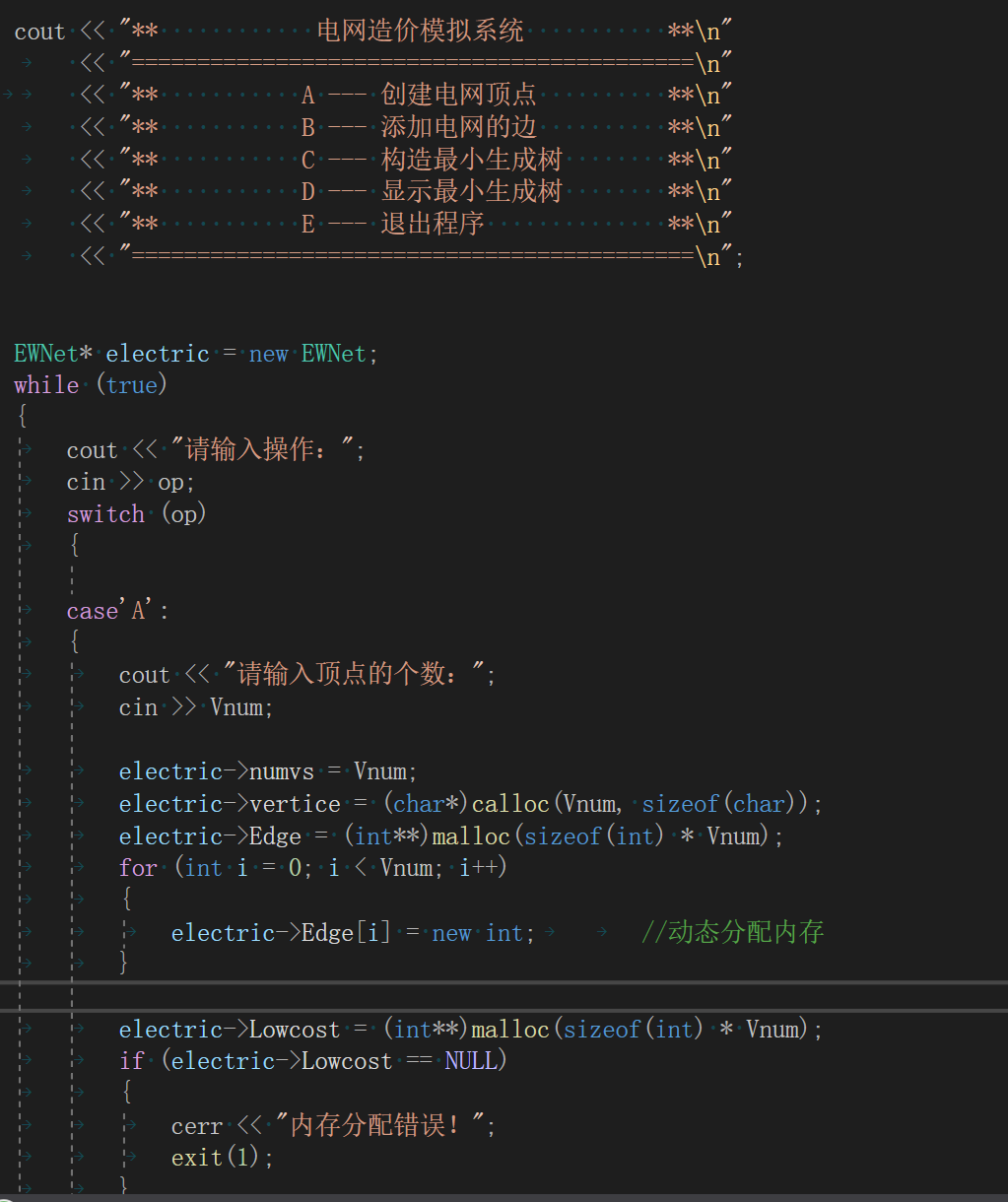
流程图：

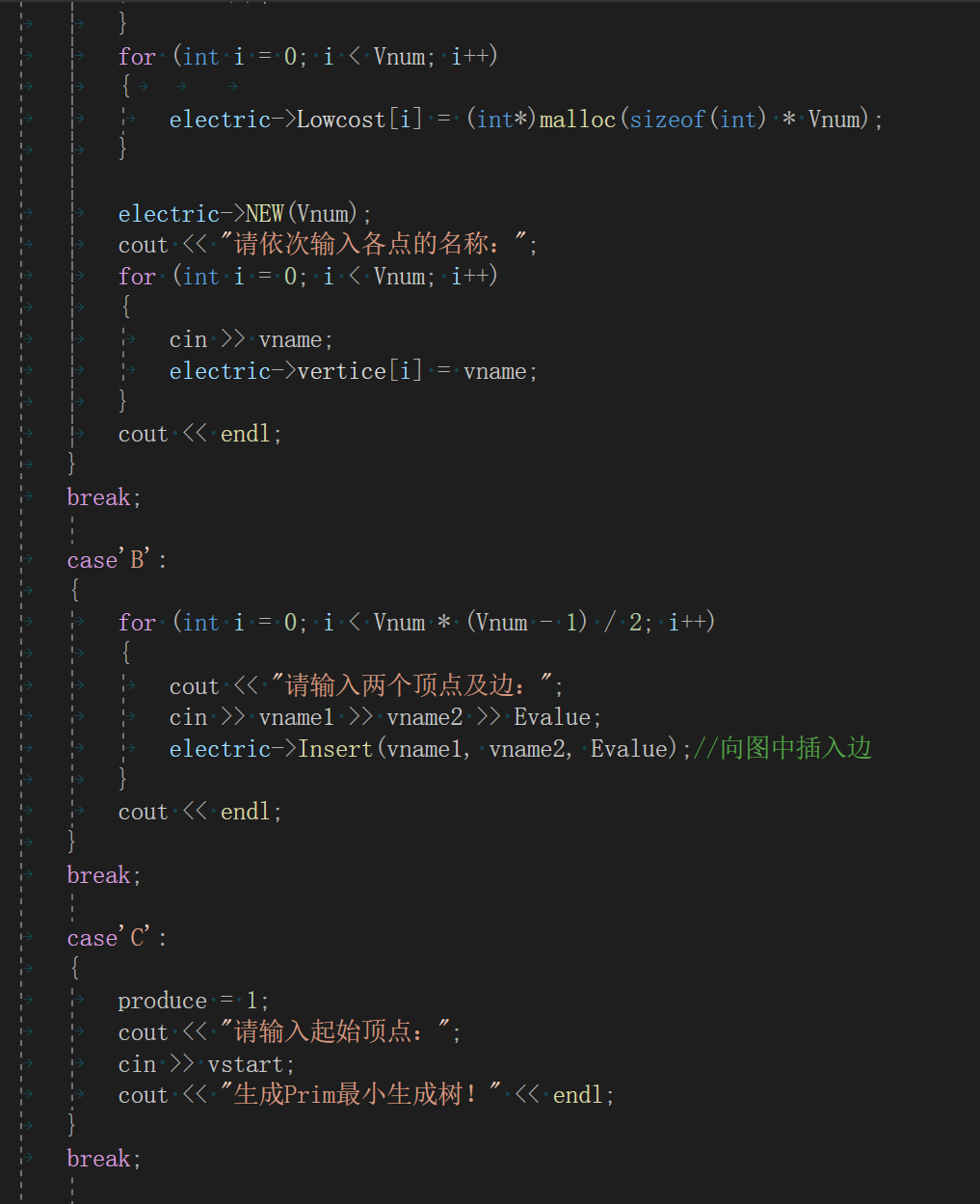


程序运行截图：



代码实现：

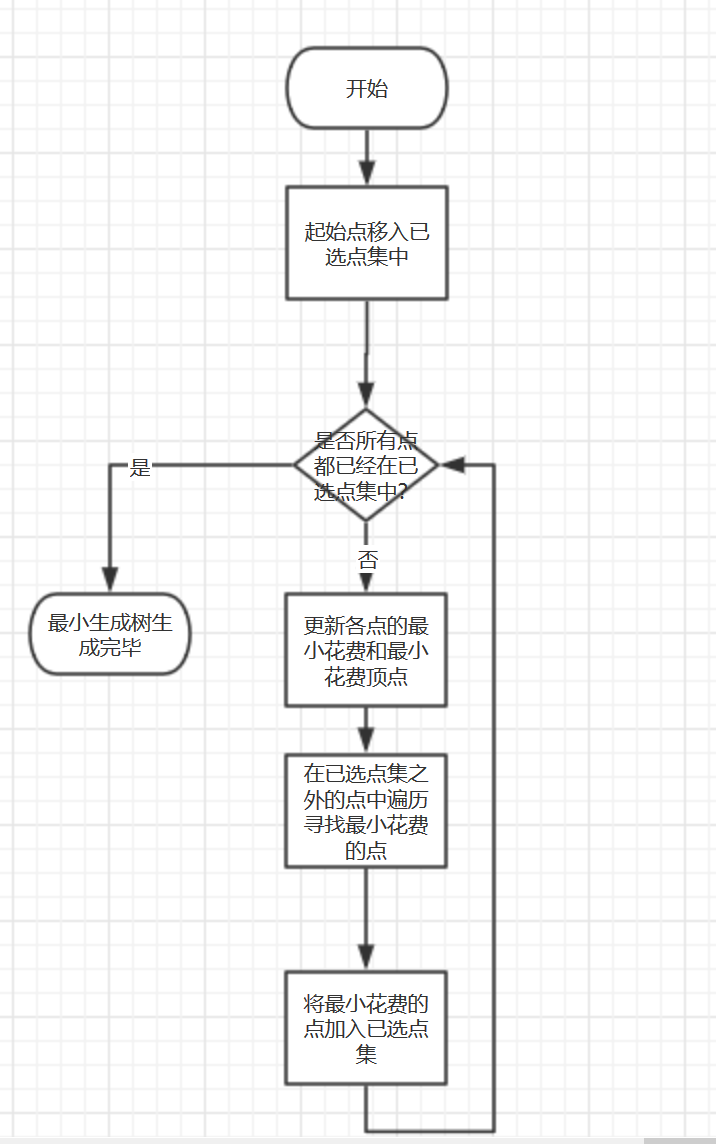




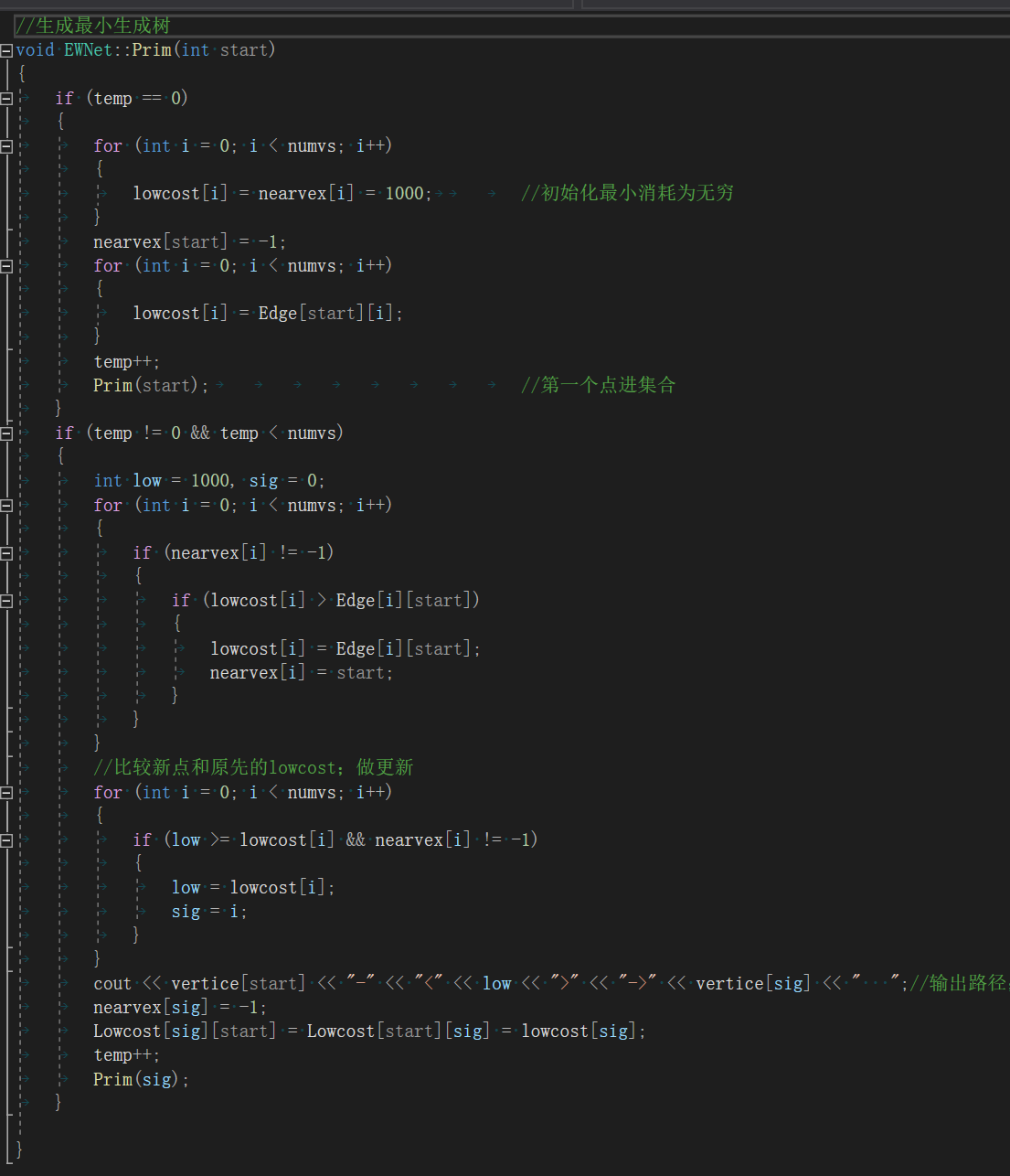


## 3.2 建立最小生成树

流程图：



代码实现：



3.3 其他成员函数的实现

void EWNet::NEW(int sz) {

for (int i = 0; i < sz; i++)

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

{

Edge[i][j] = maxsize;

Lowcost[i][j] = maxsize;

}

};

//寻找相应顶点

int EWNet::getvertice(char vs)

{

for (int i = 0; i < numvs; i++)

{

if (vertice[i] == vs)

return i;

}

return -1;

};

//输出顶点

void EWNet::OutputVertice()

{

for (int i = 0; i < numvs; i++)

{

cout << vertice[i] << " ";

}

}

//输出边

void EWNet::OutputEdge()

{

for (int i = 0; i < numvs; i++)

{

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

{

cout << Edge[i][j] << " ";

if (j == numvs - 1)cout << "\n";

}

}

};

//输出最小花费

void EWNet::OutputLowcost()

{

for (int i = 0; i < numvs; i++)

{

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

{

cout << Lowcost[i][j] << " ";

if (j == numvs - 1)cout << "\n";

}

}

};

//向图中插入边

void EWNet::Insert(char v1, char v2, int value)

{

int v1pos = getvertice(v1);

int v2pos = getvertice(v2);

Edge[v1pos][v2pos] = Edge[v2pos][v1pos] = value;

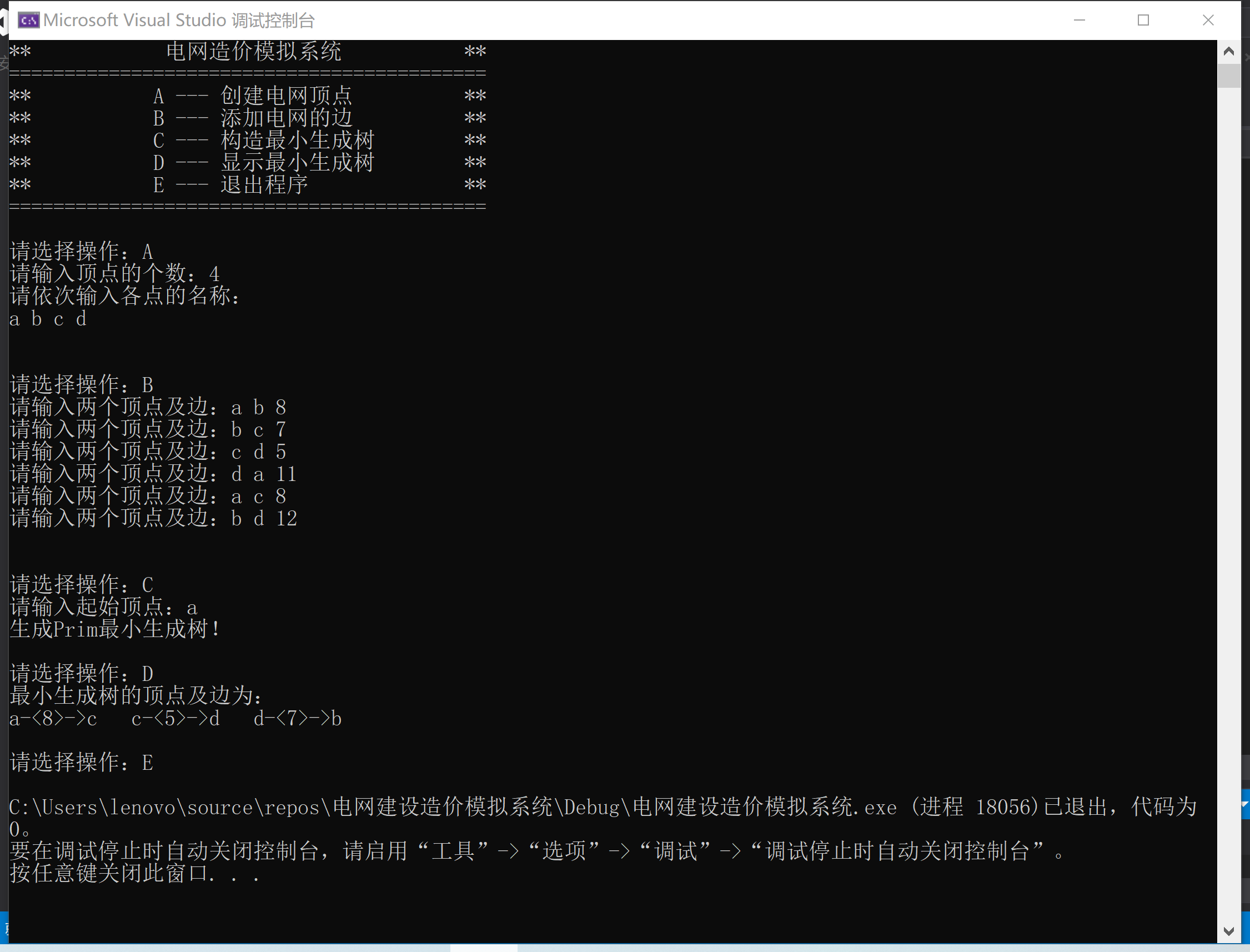
};

# 4 测试

## 4.1 功能测试

### 4.1.1 正常测试

**实验结果：**

****

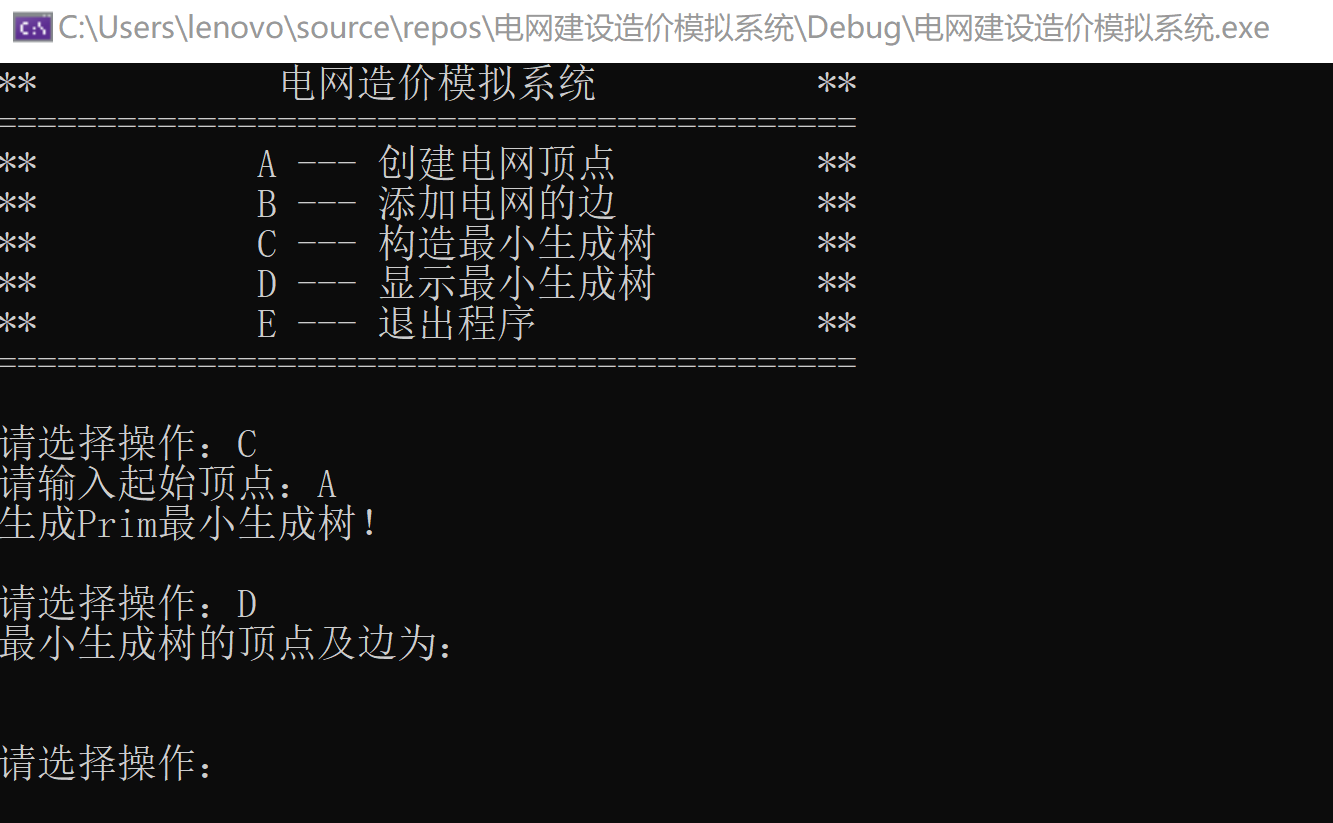
## 4.2 边界测试

### 4.2.1 空图的最小生成树

**预期结果：**

程序正常运行不崩溃

**实验结果：**



## 4.3 出错测试

### 4.3.1 未生成最小生成树就要求显示

预测结果：程序给出提示，不崩溃。



### 4.3.2 未知指令

预期结果：

程序正常运行不崩溃。

运行结果：

