**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： 张梓瀚

学 号： 2051943

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 项目分析 - 3 -](#_Toc121493401)

[1.1项目背景 - 3 -](#_Toc121493402)

[1.2 项目要求 - 3 -](#_Toc121493403)

[1.2.1 功能要求 - 3 -](#_Toc121493404)

[1.2.2 输入格式 - 3 -](#_Toc121493405)

[1.2.3 输出格式 - 3 -](#_Toc121493406)

[1.2.4 项目示例 - 4 -](#_Toc121493407)

[2 项目设计 - 4 -](#_Toc121493408)

[2.1 数据结构设计 - 4 -](#_Toc121493409)

[2.2 类设计 - 4 -](#_Toc121493410)

[2.2.1 边类（edge） - 4 -](#_Toc121493411)

[2.3 电网建设造价模拟系统算法 - 5 -](#_Toc121493412)

[2.3.1 主要思想 - 5 -](#_Toc121493413)

[2.3.2代码 - 5 -](#_Toc121493414)

[2.3 流程图 - 7 -](#_Toc121493415)

[3 项目测试 - 9 -](#_Toc121493416)

[3.1 正常情况 - 9 -](#_Toc121493417)

[3.2 所有点形成环的情况 - 9 -](#_Toc121493418)

[3.3 只有两个点的情况 - 10 -](#_Toc121493419)

[3.4 所给无向图本身是一棵树 - 10 -](#_Toc121493420)

[3.5 所给图不连通 - 11 -](#_Toc121493421)

# 1 项目分析

## 1.1项目背景

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 项目要求

### 1.2.1 功能要求

在每个小区之间都可以设置一条电网线路，都要付出相应的经济代价。n个小区之间最多可以有n（n-1）/2条线路，选择其中的n-1条使总的耗费最少。

### 1.2.2 输入格式

（见项目示例）

### 1.2.3 输出格式

（见项目示例）

### 1.2.4 项目示例



# 2 项目设计

## 2.1 数据结构设计

按照题目要求，本题采用Prim算法构造最小生成树来解决，为了方便后续查找，采用链式前向星和邻接矩阵相结合的方式存储无向图。

## 2.2 类设计

### 2.2.1 边类（edge）

//边结构体

struct edge {

int x, y, len;

edge() {

x = 0, y = 0, len = 0;

}

bool operator <(const edge& rhs) const {

return len < rhs.len;

}

bool operator >(const edge& rhs) const {

return len > rhs.len;

}

bool operator ==(const edge& rhs) const {

return x == rhs.x && y == rhs.y && len == rhs.len;

}

edge& operator =(const edge& rhs) {

if (rhs == \*this)

return \*this;

x = rhs.x;

y = rhs.y;

len = rhs.len;

return \*this;

}

edge(const int& \_x, const int& \_y, const int& \_v) {

x = \_x;

y = \_y;

len = \_v;

}

}edges[max\_point\_num \* max\_point\_num];

## 2.3 电网建设造价模拟系统算法

### 2.3.1 主要思想

由题意可知，电网线路可以抽象成一个有n个结点的无向图，为了使电网造价最低且连通所有n个点，需要建立最小生成树，本题中采用Prim算法进行构造，用一个点集合记录已经加入到最小生成树中的顶点，用lowcost数组记录未加入集合的顶点到集合的最短距离，closest数组记录到当前点最近的顶点，每次选择lowcost最小的顶点加入点集，不断更新这两个数组，直到所有顶点加入点集，就完成了最小生成树的构建。

### 2.3.2代码

void Prim(int st) {

//初始化操作，获取基本信息。

for (int i = 1; i <= n; i++) {

if (i == st)

lowcost[i] = 0;

else

lowcost[i] = g[st][i];

closest[i] = st;

}

int minn, pos;//记录距离集合最近的点，minn代表距离，pos代表该点的终边下标。

sum = 0;

for (int i = 1; i <= n; i++) {

minn = INF;

for (int j = 1; j <= n; j++) {

//找出距离点集合最近的边。

if (lowcost[j] != 0 && lowcost[j] < minn) {

minn = lowcost[j];

pos = j;

}

}

if (minn == INF)

break;

sum += minn;//计算最小生成树的权值之和。

lowcost[pos] = 0;//加入点集合。

for (int j = 1; j <= n; j++) {

//由于点集合中加入了新的点，我们要去更新。

if (lowcost[j] != 0 && g[pos][j] < lowcost[j]) {

lowcost[j] = g[pos][j];

closest[j] = pos;//改变与顶点j相连的顶点序号。

}

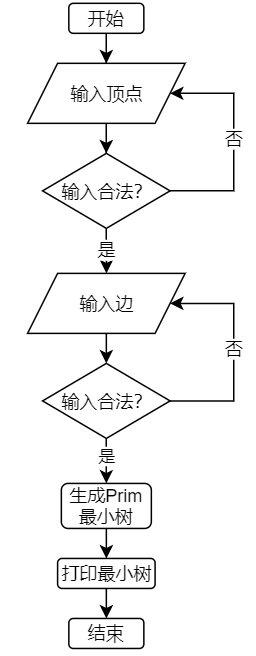
}

}

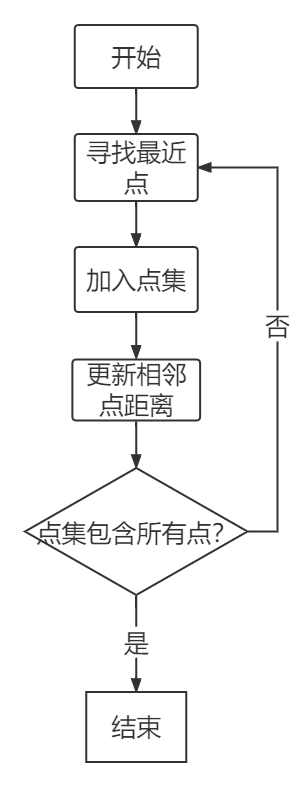
}

### 2.3 流程图

#### 2.3.1 项目主体功能流程图



#### 2.3.2 Prim最小生成树流程图



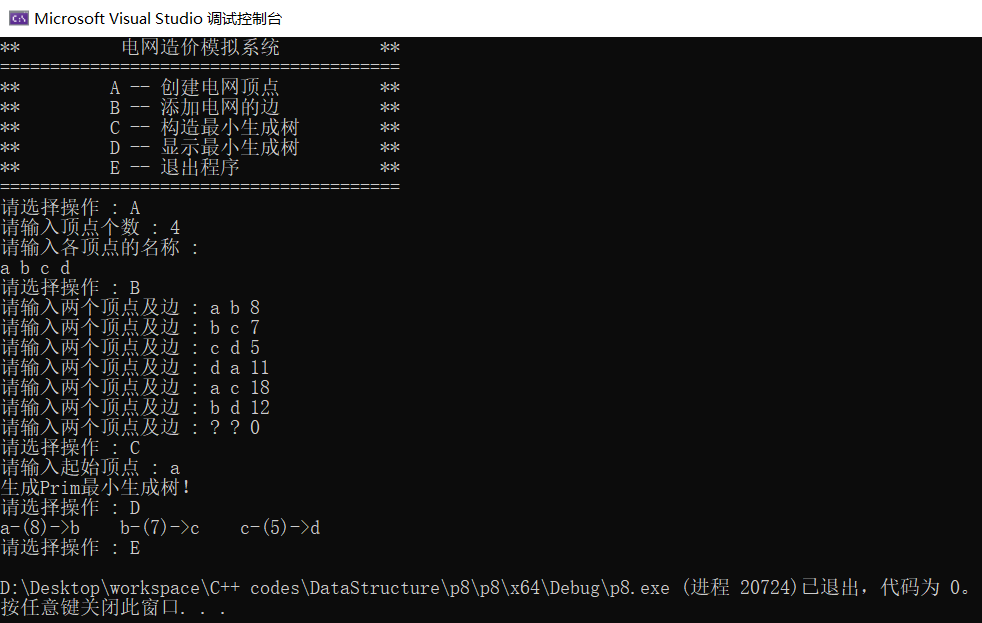
# 3 项目测试

## 3.1 正常情况

测试用例：

预期结果：

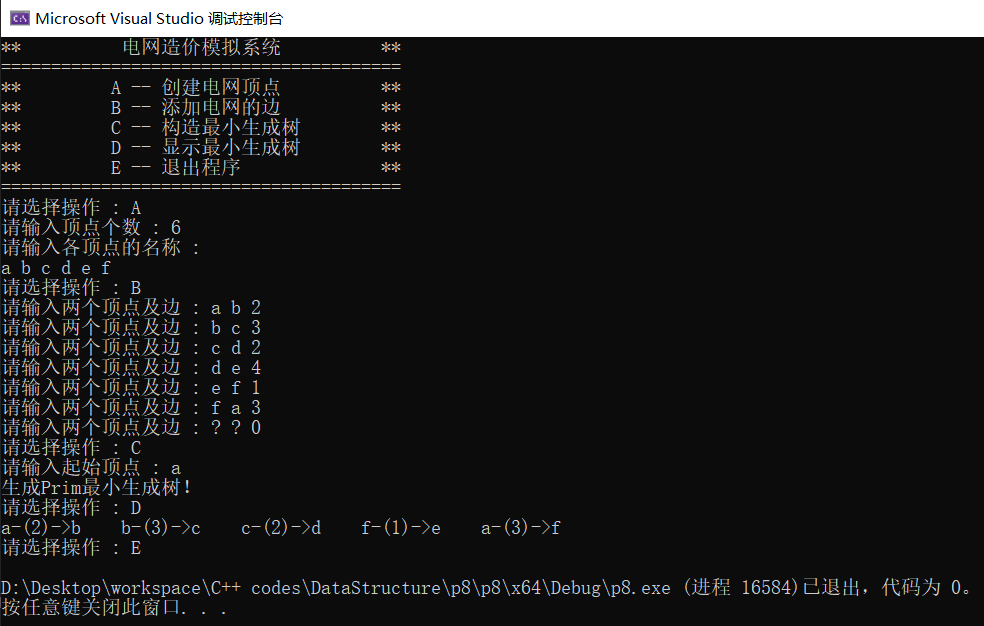
实验结果：



## 3.2 所有点形成环的情况

测试用例：

预期结果：

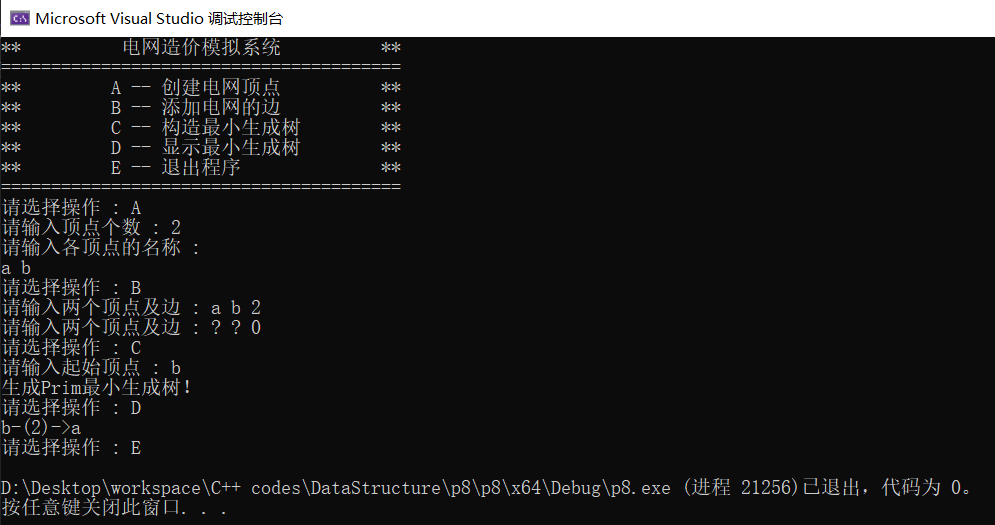
实验结果：

## 3.3 只有两个点的情况

测试用例：

预期结果：

实验结果：

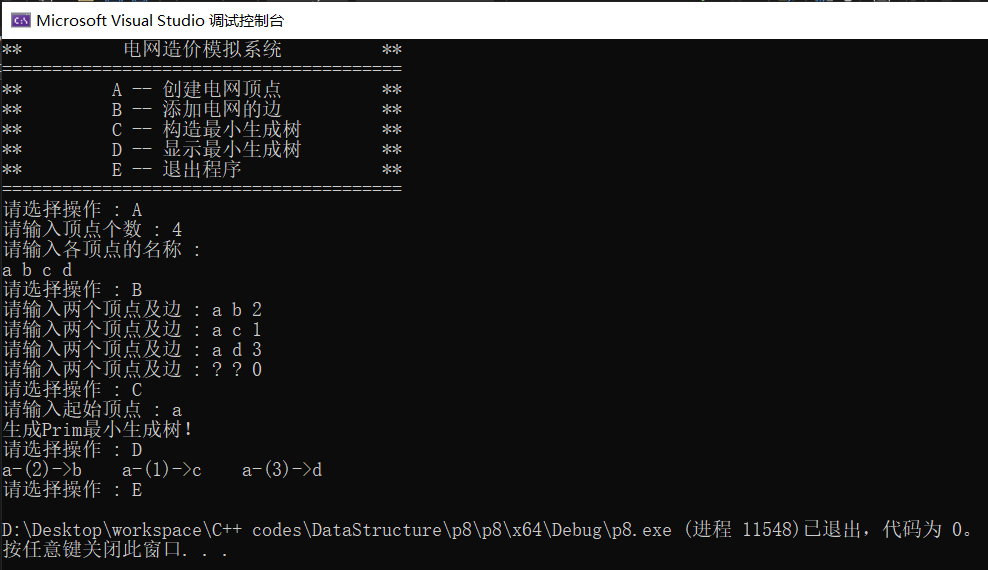


## 3.4 所给无向图本身是一棵树

测试用例：

预期结果：

实验结果：



## 3.5 所给图不连通

测试用例：

预期结果：

实验结果：

