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：安江涛

学　　号：1952560

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

1.1 项目简介

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。

项目功能要求：

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。

2 设计

2.1 数据结构设计

题目要求选择其中 n-1 条线路使总的耗费最少，是典型的最小生成树，最小生成树的方法有 prim 和 kruskal 两种方法，本文档是用的是 prim 方法，另一种方法只提供代码。

2.2 类结构设计

本项目含有一个类，System 类，实现包括电网造价模拟系统的初始化、创建顶点、添加边、构造最小生成树、显示最小生成树等功能。

2.3 成员与操作设计

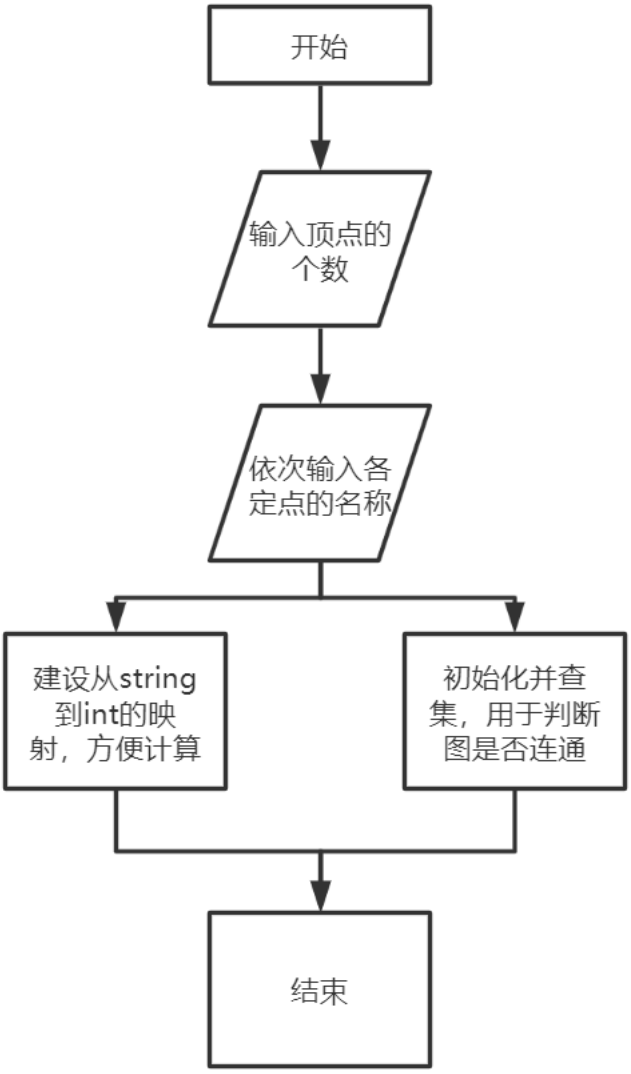
System 类

```
class System {
public:
    std::string find(std::string x) {
        return _par[x] == x ? x : _par[x] = find(_par[x]);
    }
    void prim(std::string str);
    void initializeVertex();
    void addEdge();
    void print();
    bool exist(std::string str);
private:
    int _size=0;
    bool _flag;//是否进行了操作C
    bool _exist;//是否存在最小生成树
    std::vector<std::string> _vertex;//顶点集
    std::vector<edge> _path;//存路径
    std::unordered_map<std::string, int> _map;//映射
    int _graph[maxn][maxn];
    bool _visit[maxn];//判断顶点有没有访问过
    std::map<std::string, std::string> _par;
};
```

3 实现

3.1 创建电网顶点

3.1.1 流程图



3.1.2 核心代码

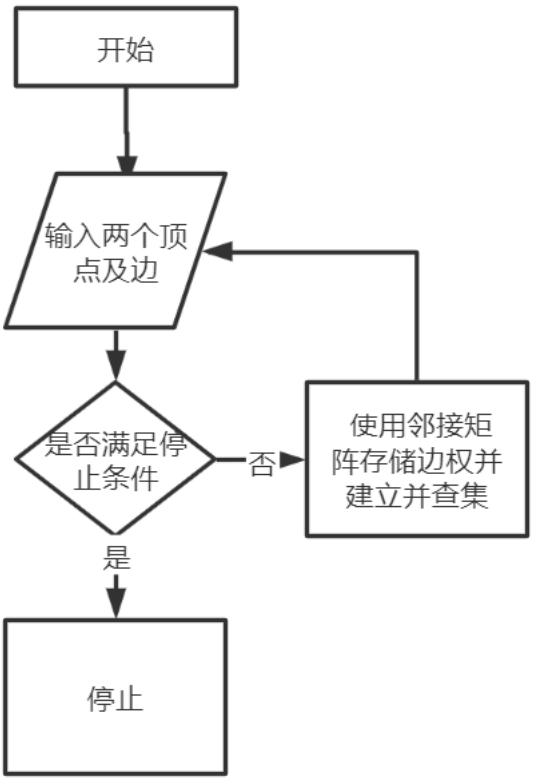
```
void System::initializeVertex() {
    _par.clear();
    _flag=false;
    _exist=false;
    _vertex.clear();//清空顶点
    _map.clear();//清空映射
    std::cout << "Please enter the number of vertices：";
    int num;
    std::string str;
    std::cin >> num;
    while(!cinClear()||num<2){
        if(num<2)
            std::cout<<"Input errors, please re-enter:";
        std::cin>>num;
    }
    _size=num;
    std::cout << "Please enter the name of each vertex in turn：" << '\n';
    for (int i = 0; i < num; i++) {
        std::cin >> str;
        while(!cinClear())
            std::cin>>str;
        _vertex.push_back(str);
        _map[str] = i;
        _par[str]=str;
    }
    for(int i=0;i<_size;i++)
        for(int j=0;j<_size;j++)
            _graph[i][j]=INF;
    std::cout << '\n';
}
```

3.1.3 截屏示例

```
root@iZbp180dvecytxjnhuzrdnZ:~/css_design/subject# g++ p8_prim.cpp -o p8.out
root@iZbp180dvecytxjnhuzrdnZ:~/css_design/subject# ./p8.out
**         Power grid cost simulation system        **
========================================================
**          A---Create grid vertex                  **
**          B---Add the edge of the grid            **
**          C---Build a minimum spanning tree       **
**          D---Show minimum spanning tree          **
**          E---exit the program                    **
Please enter operation：A
Please enter the number of vertices：-3
Input errors, please re-enter:4
Please enter the name of each vertex in turn：
a b c d

Please enter operation：
```

3.2 添加电网的边

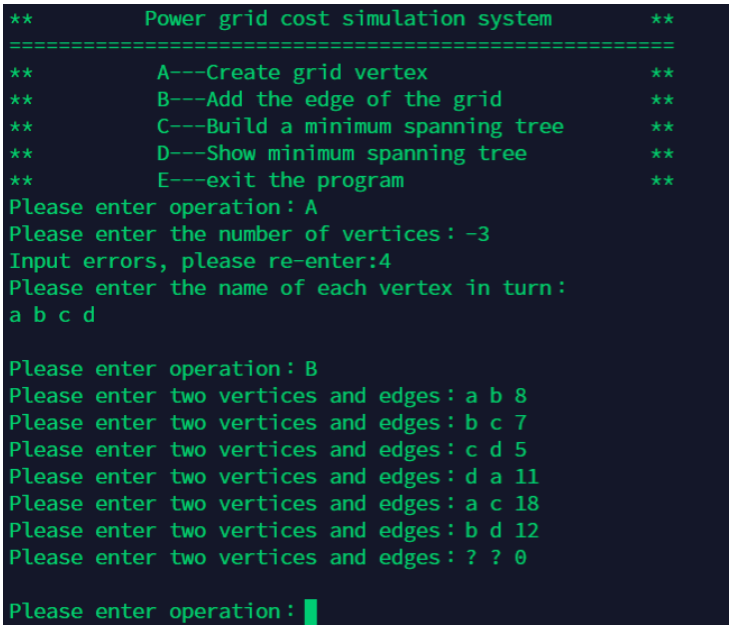
3.2.1 流程图



3.2.2 核心代码

```
void System::addEdge() {
    edge e;
    while (true) {
        std::cout << "Please enter two vertices and edges：";
        std::cin >> e.from >> e.to >> e.dist;
        if (e.from == "?" && e.to == "?" && e.dist == 0)
            break;
        _graph[_map[e.from]][_map[e.to]]=e.dist;
        _graph[_map[e.to]][_map[e.from]]=e.dist;
        _par[find(e.from)]=find(e.to);
    }
    std::cout << '\n';
}
```

3.2.3 截屏示例



3.3 构造最小生成树

3.3.1 算法描述

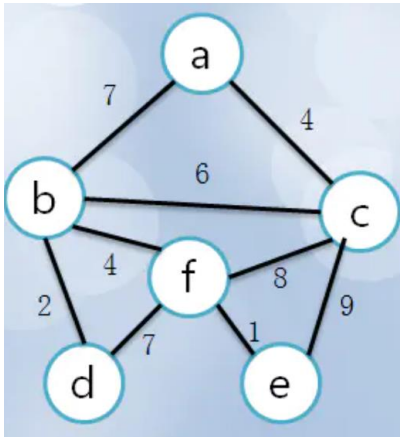
- 在一个加权连通图中，顶点集合 V，边集合为 E

- 任意选出一个点作为初始顶点,标记为 visit,计算所有与之相连接的点的距离，选择距离最短的，标记 visit.
- 重复以下操作，直到所有点都被标记为 visit：

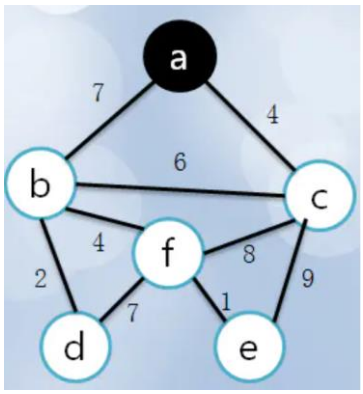
在剩下的点中，计算与已标记 visit 点距离最小的点，标记 visit,证明加入了最小生成树。

下面我们来看一个最小生成树生成的过程：

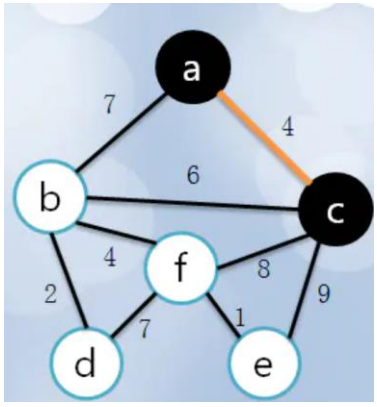
1. 起初，从顶点 a 开始生成最小生成树



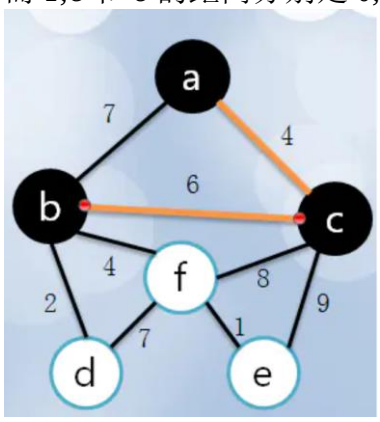
2.选择顶点 a 后，顶点啊置成 visit（涂黑）,计算周围与它连接的点的距离：



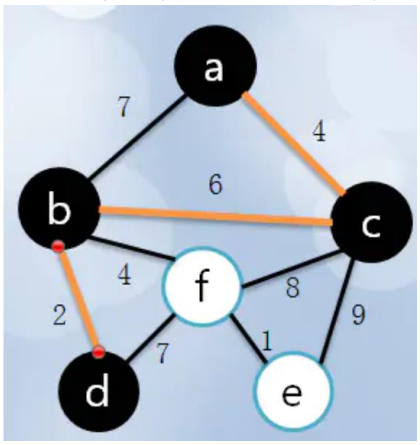
3.与之相连的点距离分别为 7,6,4，选择 c 点距离最短，涂黑 c，同时将这条边高亮加入最小生成树：



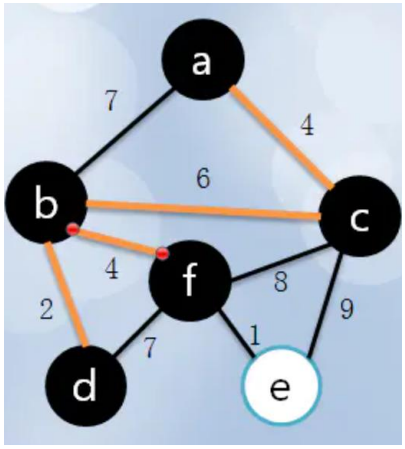
4. 计算与 a,c 相连的点的距离（已经涂黑的点不计算），因为与 a 相连的已经计算过了，只需要计算与 c 相连的点，如果一个点与 a,c 都相连，那么它与 a 的距离之前已经计算过了，如果它与 c 的距离更近，则更新距离值，这里计算的是未涂黑的点距离涂黑的点的最近距离，很明显，b 和 a 为 7，b 和 c 的距离为 6，更新 b 和已访问的点集距离为 6，而 f,e 和 c 的距离分别是 8,9，所以还是涂黑 b,高亮边 bc：



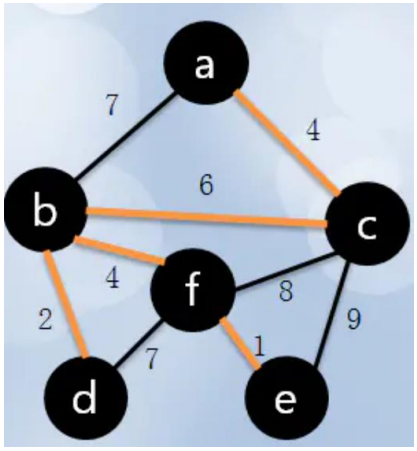
5.接下来很明显，d 距离 b 最短，将 d 涂黑，bd 高亮：



6.f 距离 d 为 7，距离 b 为 4，更新它的最短距离值是 4，所以涂黑 f，高亮 bf：



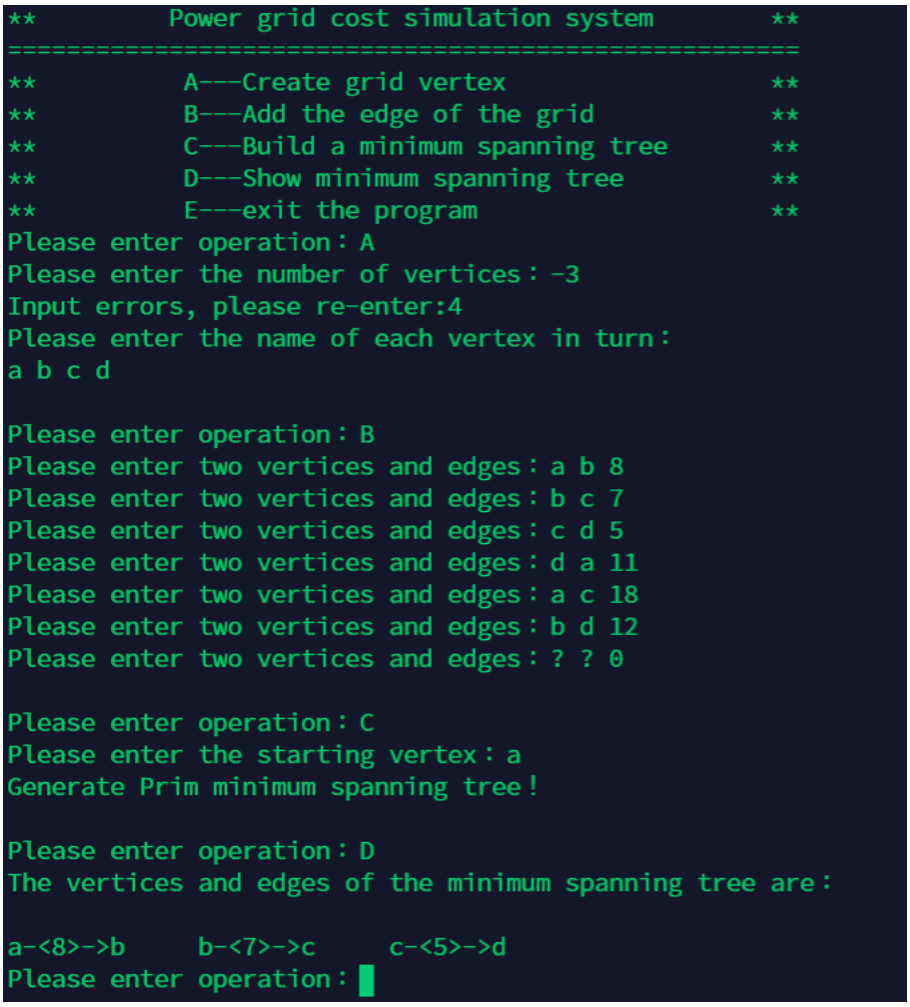
7.最后只有 e 了：



3.3.2 核心代码

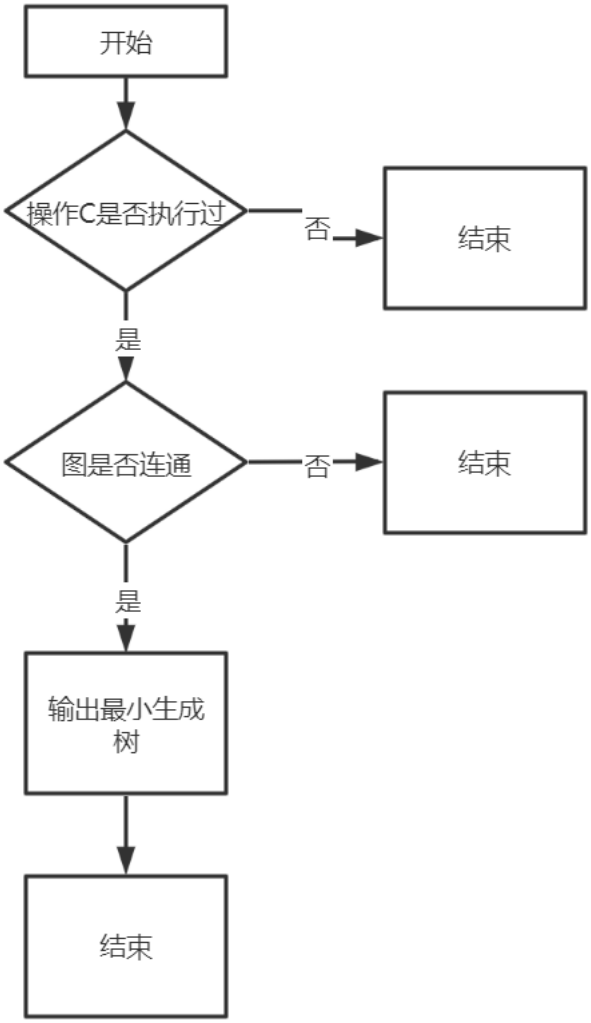
```
void System::prim(std::string str) {
    _flag=true;
    for(int i=0;i<_size-1;i++){
        if(find(_vertex[i])!=find(_vertex[i+1])){
            _exist=false;
            return;
        }
    }
    int cur=_map[str];
    _path.clear();      for(int i=0;i<_size;i++)
        _visit[i]=false;
    _visit[cur]=false;
    int index=cur;
    int dist[maxn]={0};
    for(int i=0;i<_size;i++)
        dist[i]=_graph[cur][i];
    for(int i=1;i<_size;i++){
        int minor=INF;
        for(int j=0;j<_size;j++){
            if(!_visit[j]&&dist[j]<minor){
                minor=dist[j];
                index=j;
            }
        }
        _visit[index]=true;
        _path.push_back(edge(_vertex[cur],_vertex[index],minor));
        for(int j=0;j<_size;j++){
            if(!_visit[j]&&dist[j]>_graph[index][j])
                dist[j]=_graph[index][j];
        }
        cur=index;
    }
    _exist=true;
}
```

3.3.3 截屏示例


** Power grid cost simulation system **
** A---Create grid vertex **
** B---Add the edge of the grid **
** C---Build a minimum spanning tree **
** D---Show minimum spanning tree **
** E---exit the program **
Please enter operation : A
Please enter the number of vertices : -3
Input errors, please re-enter:4
Please enter the name of each vertex in turn :
a b c d
Please enter operation : B
Please enter two vertices and edges : a b 8
Please enter two vertices and edges : b c 7
Please enter two vertices and edges : c d 5
Please enter two vertices and edges : d a 11
Please enter two vertices and edges : a c 18
Please enter two vertices and edges : b d 12
Please enter two vertices and edges : ? ? 0
Please enter operation : C
Please enter the starting vertex : a
Generate Prim minimum spanning tree !
Please enter operation : D
The vertices and edges of the minimum spanning tree are :
a-<8>->b b-<7>->c c-<5>->d
Please enter operation :

3.4 显示最小生成树

3.4.1 流程图



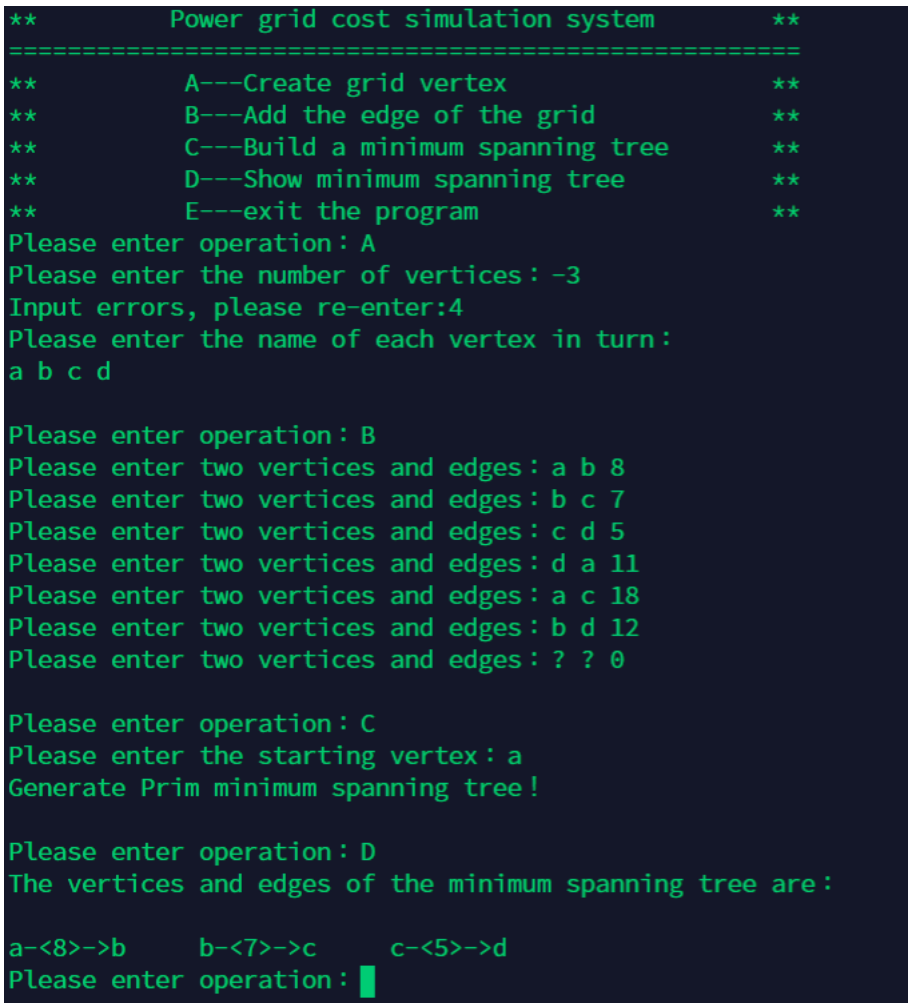
3.4.2 核心代码

```
void System::print() {
    if(!_flag){
        std::cout<<"Please enter operation C to generate a minimum spanning tree!"<<'\n';
        return;
    }
    if(!_exist){
        std::cout<<"No minimum spanning tree!"<<'\n';
        return;
    }
    std::cout << "The vertices and edges of the minimum spanning tree are                    :
" << '\n' << '\n';
    for (int i = 0; i < _path.size(); i++) {
        std::cout << _path[i].from << "-<" << _path[i].dist << ">->" << _path[i].to;
        std::cout << "     ";
    }
    std::cout << '\n';
}
```

3.4.3 截屏示例

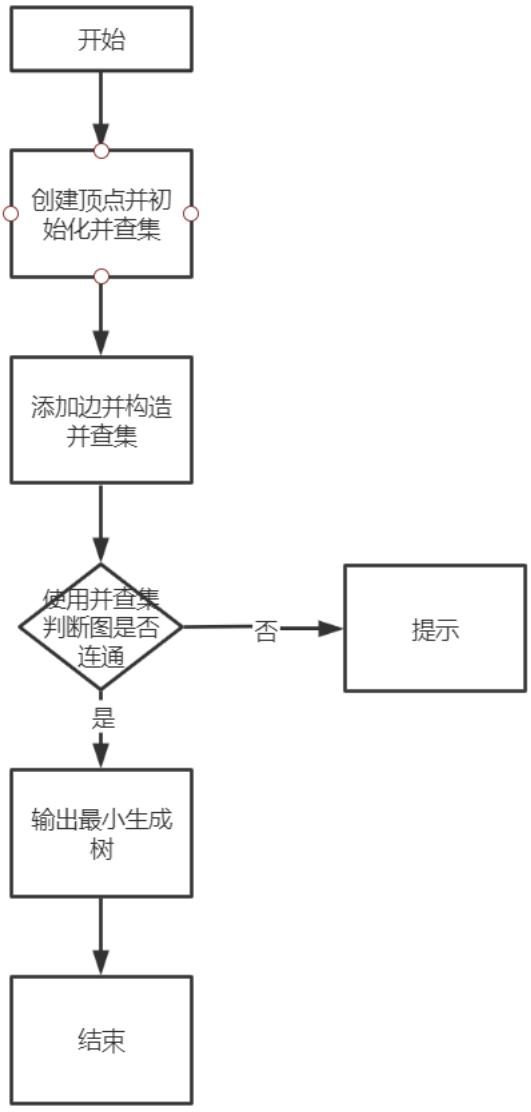
```
**          Power grid cost simulation system          **
=======================================================
**           A---Create grid vertex                    **
**           B---Add the edge of the grid              **
**           C---Build a minimum spanning tree         **
**           D---Show minimum spanning tree            **
**           E---exit the program                      **
Please enter operation：a
Input error!
Please enter operation：A
Please enter the number of vertices：4
Please enter the name of each vertex in turn：
a b c d

Please enter operation：C
Please enter the starting vertex：a
Generate Prim minimum spanning tree！

Please enter operation：D
No minimum spanning tree!
Please enter operation：
```

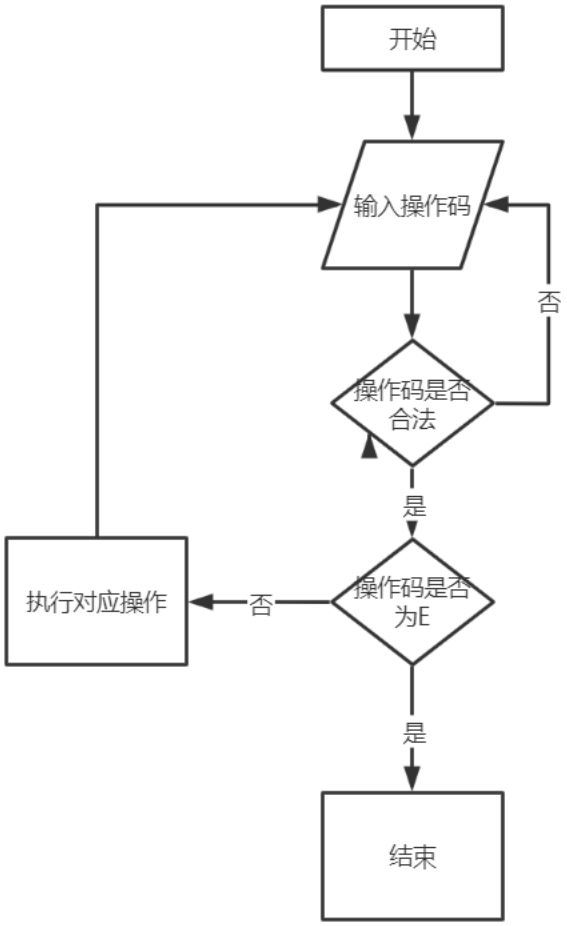
```
**          Power grid cost simulation system          **
=======================================================
**           A---Create grid vertex                    **
**           B---Add the edge of the grid              **
**           C---Build a minimum spanning tree         **
**           D---Show minimum spanning tree            **
**           E---exit the program                      **
Please enter operation：A
Please enter the number of vertices：4
Please enter the name of each vertex in turn：
a b c d

Please enter operation：D
Please enter operation C to generate a minimum spanning tree!
Please enter operation：
```


** Power grid cost simulation system **
** A---Create grid vertex **
** B---Add the edge of the grid **
** C---Build a minimum spanning tree **
** D---Show minimum spanning tree **
** E---exit the program **
Please enter operation : A
Please enter the number of vertices : -3
Input errors, please re-enter:4
Please enter the name of each vertex in turn :
a b c d
Please enter operation : B
Please enter two vertices and edges : a b 8
Please enter two vertices and edges : b c 7
Please enter two vertices and edges : c d 5
Please enter two vertices and edges : d a 11
Please enter two vertices and edges : a c 18
Please enter two vertices and edges : b d 12
Please enter two vertices and edges : ? ? 0
Please enter operation : C
Please enter the starting vertex : a
Generate Prim minimum spanning tree !
Please enter operation : D
The vertices and edges of the minimum spanning tree are :
a-<8>->b b-<7>->c c-<5>->d
Please enter operation :

3.6 总体系统的实现

3.61 总体系统流程图





3.6.2 总体系统核心代码

```
void solve() {
    std::string str[7];
    str[0]="**        Power grid cost simulation system       **";
    str[1]="==========================================================
==";
    str[2]="**          A---Create grid vertex                **";
    str[3]="**          B---Add the edge of the grid          **";
    str[4]="**          C---Build a minimum spanning tree     **";
    str[5]="**          D---Show minimum spanning tree        **";
    str[6]="**          E---exit the program                  **";
    for(int i=0;i<7;i++)
        std::cout<<str[i]<<'\n';
    System sys;
    std::string num;
    while (true) {
        std::cout << "Please enter operation：  ";
        std::cin >> num;
        while(!cinClear())
            std::cin>>num;
        if (num == "A") {
            sys.initializeVertex();
        }
        else if (num == "B") {
            sys.addEdge();
        }
        else if (num == "C") {
            std::cout << "Please enter the starting vertex：  ";
            std::string str;
            std::cin >> str;
            while(!sys.exist(str)){
                std::cout<<"The grid vertex set does not exist"<<str<<",please enter again:";
                std::cin>>str;
            }
            sys.prim(str);
            std::cout << "Generate Prim minimum spanning tree！ " << '\n' << '\n';
        }
        else if (num == "D") {
            sys.print();
        }
        else if (num == "E")
            break;
        else
            std::cout<<"Input error!"<<'\n';
    }
```

3.6.3 总体系统截屏示例

```
**          Power grid cost simulation system          **
=======================================================
**           A---Create grid vertex                    **
**           B---Add the edge of the grid              **
**           C---Build a minimum spanning tree         **
**           D---Show minimum spanning tree            **
**           E---exit the program                      **
Please enter operation：A
Please enter the number of vertices：-3
Input errors, please re-enter:4
Please enter the name of each vertex in turn：
a b c d

Please enter operation：B
Please enter two vertices and edges：a b 8
Please enter two vertices and edges：b c 7
Please enter two vertices and edges：c d 5
Please enter two vertices and edges：d a 11
Please enter two vertices and edges：a c 18
Please enter two vertices and edges：b d 12
Please enter two vertices and edges：? ? 0

Please enter operation：C
Please enter the starting vertex：a
Generate Prim minimum spanning tree!

Please enter operation：D
The vertices and edges of the minimum spanning tree are：

a-<8>->b     b-<7>->c     c-<5>->d
Please enter operation：
```