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](#_Toc495668153)

[1.1 项目简介 1](#_Toc495668154)

[2 设计 1](#_Toc495668156)

[2.1 数据结构设计 1](#_Toc495668157)

[2.2 类结构设计 1](#_Toc495668158)

[2.3 成员与操作设计 1](#_Toc495668159)

[3 实现 3](#_Toc495668161)

[3.1 创建电网顶点 3](#_Toc495668182)

[3.1.1 流程图 3](#_Toc495668183)

[3.1.2 核心代码 4](#_Toc495668183)

[3.1.3 截图示例 5](#_Toc495668183)

[3.2添加电网的边 5](#_Toc495668182)

[3.2.1 流程图 5](#_Toc495668183)

[3.2.2 核心代码 6](#_Toc495668183)

[3.1.3 截图示例 6](#_Toc495668183)

[3.3 构建最小生成树 6](#_Toc495668182)

[3.3.1 算法描述 6](#_Toc495668183)

[3.3.2 核心代码 10](#_Toc495668183)

[3.3.3 截图示例 11](#_Toc495668183)

[3.4 显示最小生成树 12](#_Toc495668182)

[3.4.1 流程图 12](#_Toc495668183)

[3.4.2 核心代码 13](#_Toc495668183)

[3.4.3 截图示例 13](#_Toc495668183)

[3.6 总体系统的实现 15](#_Toc495668182)

[3.6.1 总体系统流程图 15](#_Toc495668183)

[3.6.2 总体系统核心代码 16](#_Toc495668184)

[3.6.3 总体系统截屏示例 18](#_Toc495668185)

# 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

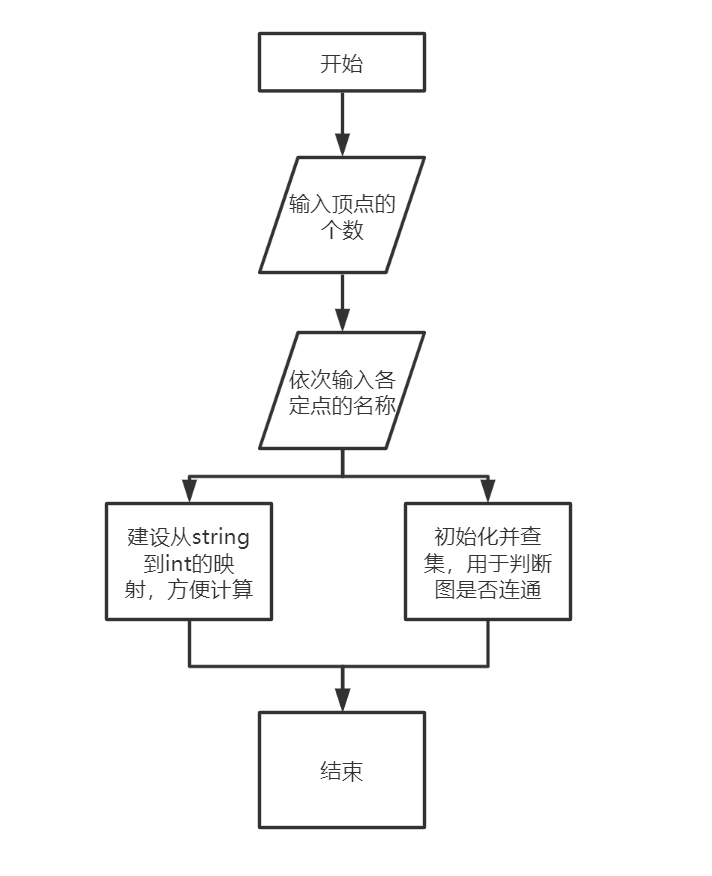
**System类**

****

# 3 实现

## 3.1 创建电网顶点

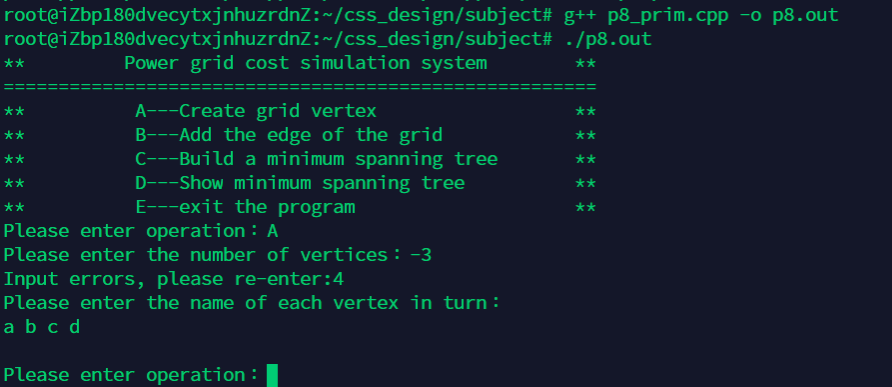
### 3.1.1 流程图



### 3.1.2 核心代码

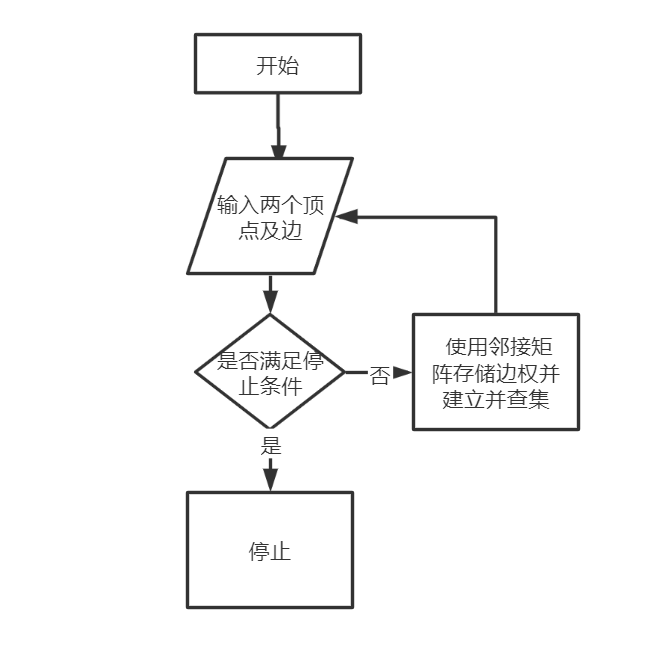


### 3.1.3 截屏示例



## 3.2 添加电网的边

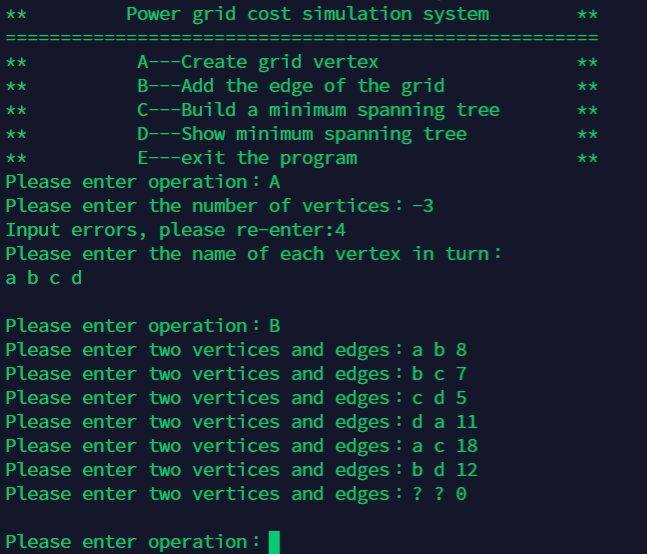
### 3.2.1 流程图



### 3.2.2 核心代码



### 3.2.3 截屏示例



## 3.3 构造最小生成树

### 3.3.1 算法描述

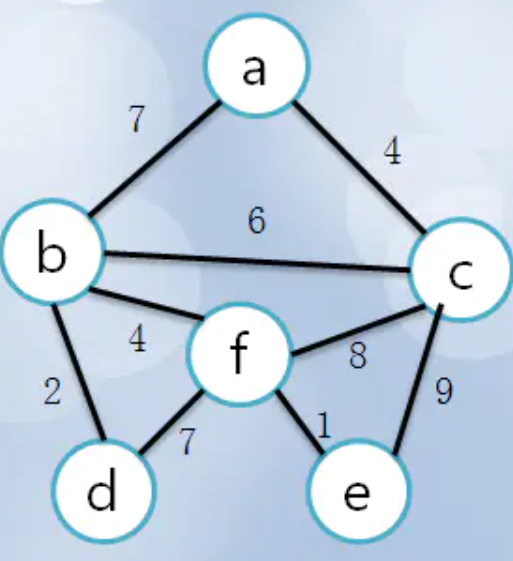
 在一个加权连通图中，顶点集合V，边集合为E

 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.

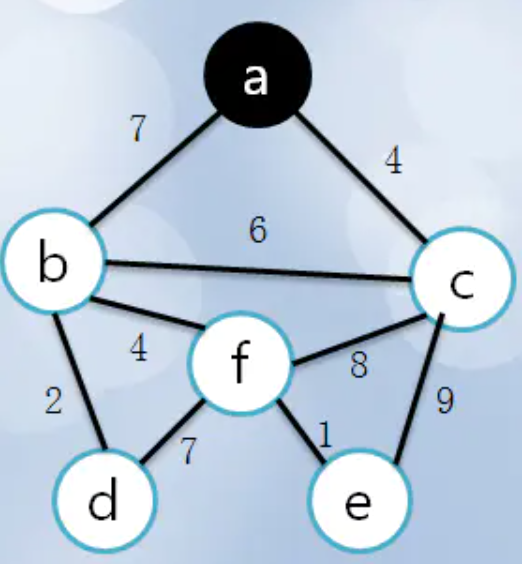
 重复以下操作，直到所有点都被标记为visit：  
在剩下的点中，计算与已标记visit点距离最小的点，标记visit,证明加入了最小生成树。

下面我们来看一个最小生成树生成的过程：

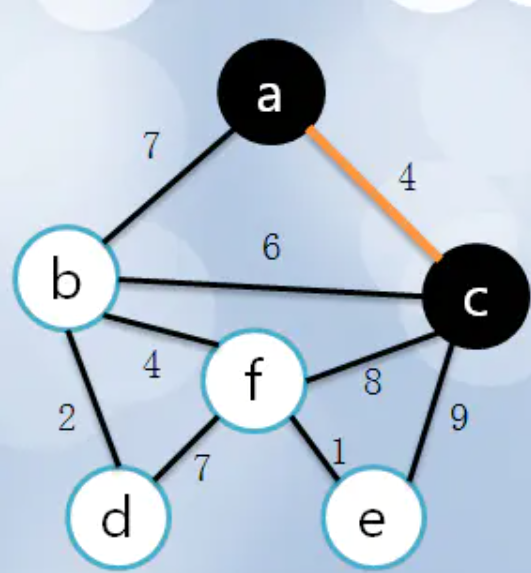
1.起初，从顶点a开始生成最小生成树



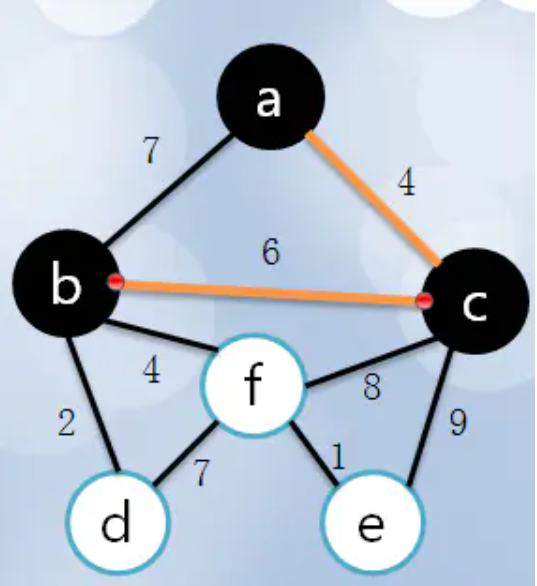
2.选择顶点a后，顶点啊置成visit（涂黑）,计算周围与它连接的点的距离：



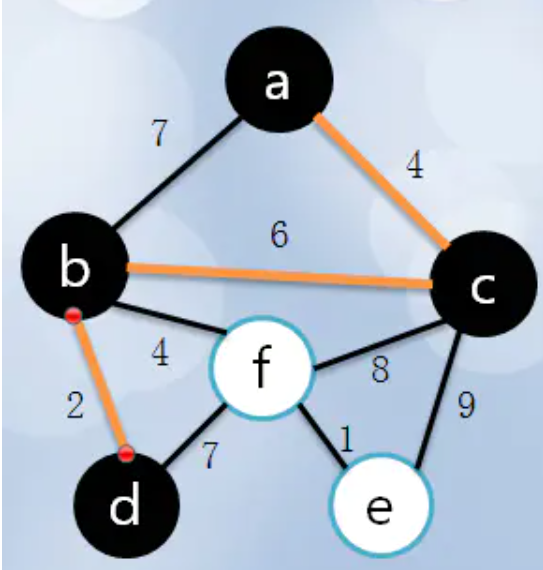
3.与之相连的点距离分别为7,6,4，选择C点距离最短，涂黑C，同时将这条边高亮加入最小生成树：



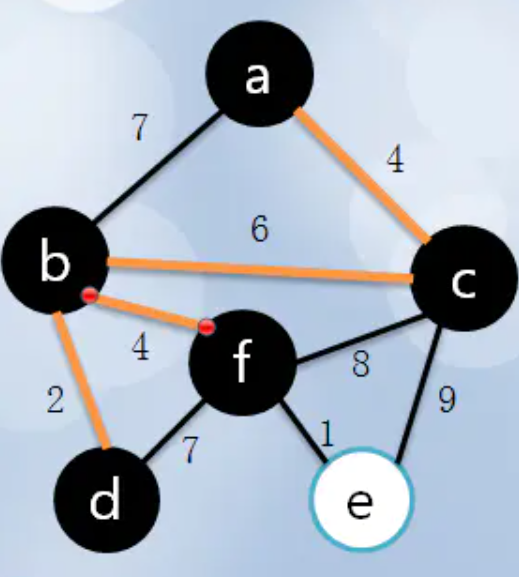
4.计算与a,c相连的点的距离（已经涂黑的点不计算），因为与a相连的已经计算过了，只需要计算与c相连的点，如果一个点与a,c都相连，那么它与a的距离之前已经计算过了，如果它与c的距离更近，则更新距离值，这里计算的是未涂黑的点距离涂黑的点的最近距离，很明显，b和a为7，b和c的距离为6，更新b和已访问的点集距离为6，而f,e和c的距离分别是8,9，所以还是涂黑b,高亮边bc：



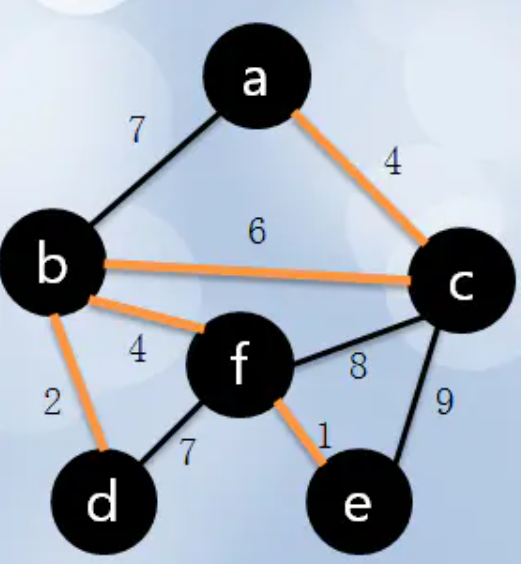
5.接下来很明显，d距离b最短，将d涂黑，bd高亮：



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：



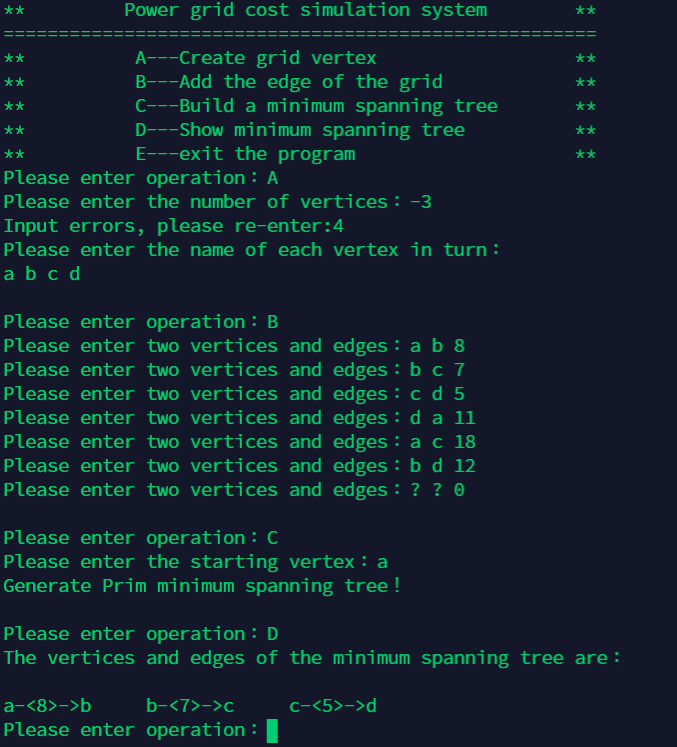
7.最后只有e了：



### 3.3.2 核心代码

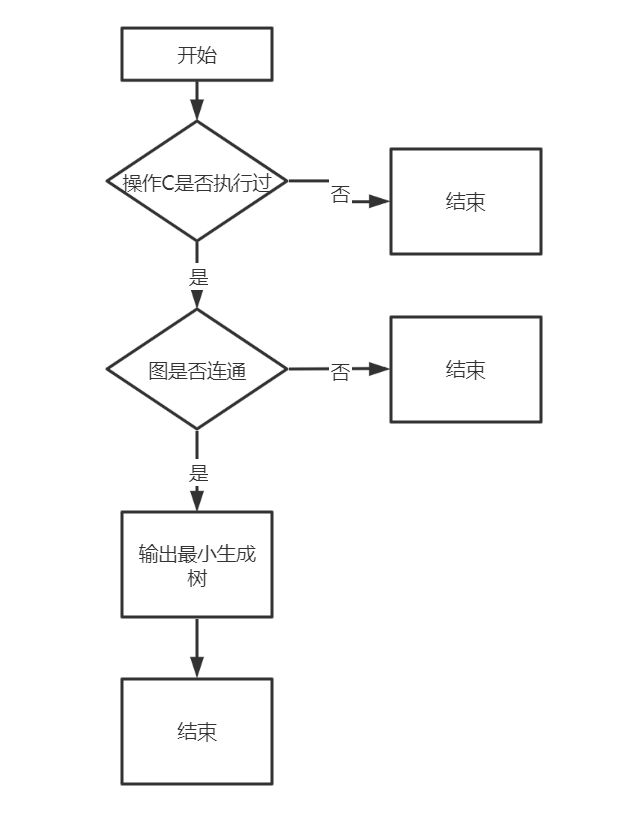


### 3.3.3 截屏示例



## 3.4 显示最小生成树

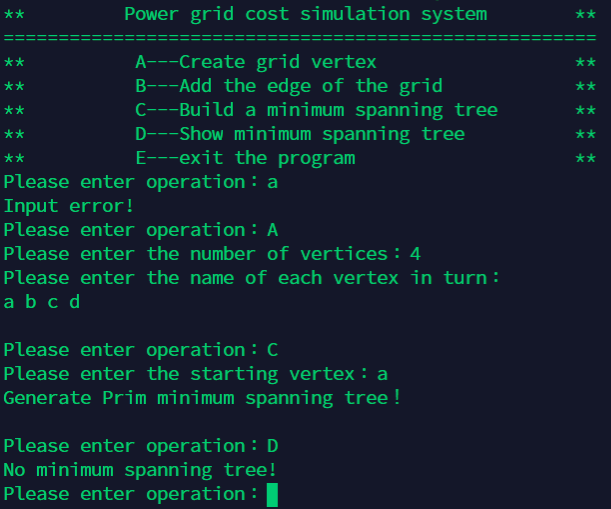
### 3.4.1 流程图

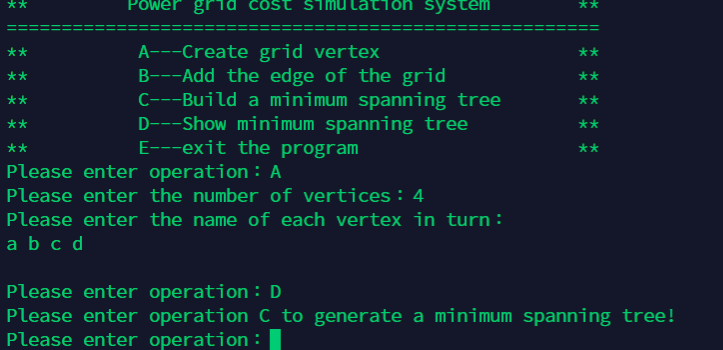


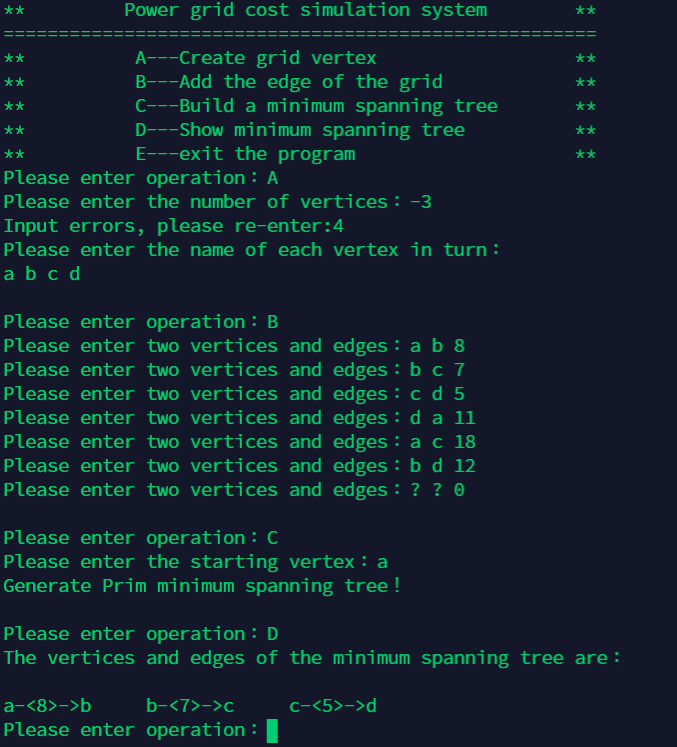
### 3.4.2 核心代码



### 3.4.3 截屏示例

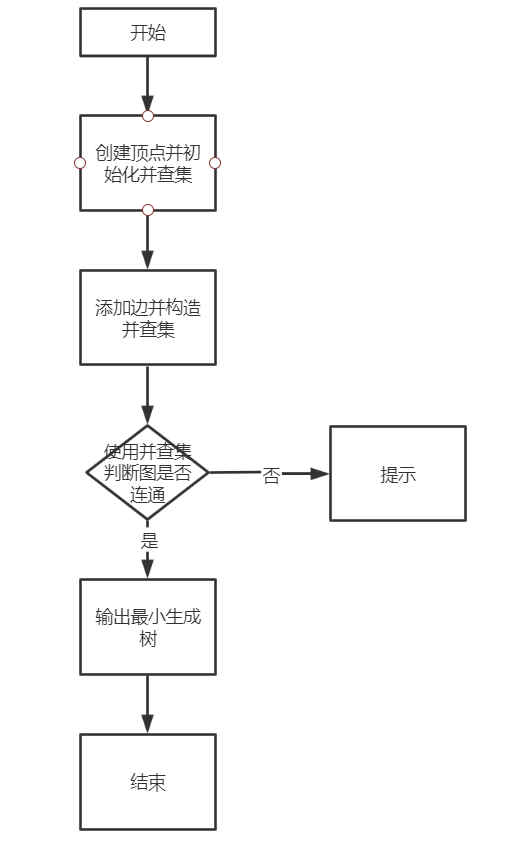


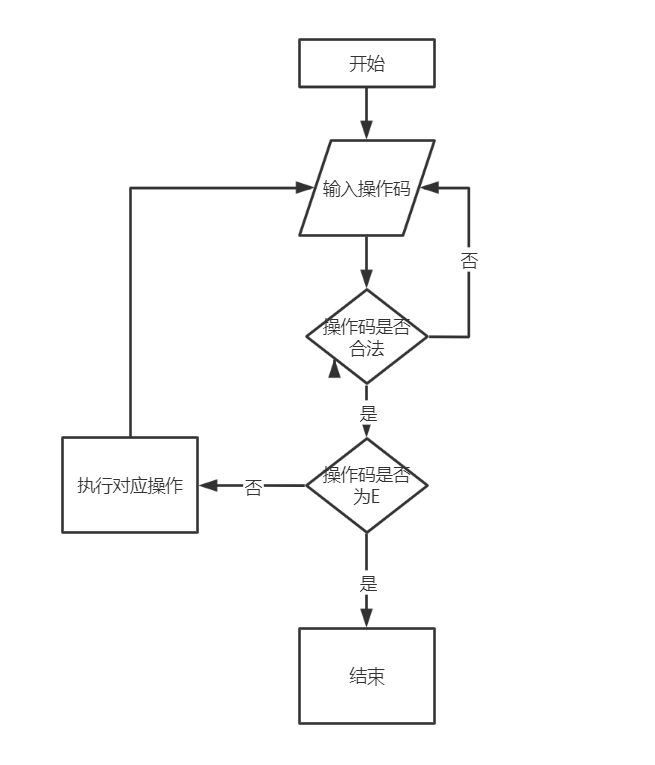




## 3.6 总体系统的实现

### 3.61 总体系统流程图





### 3.6.2 总体系统核心代码



### 3.6.3 总体系统截屏示例

