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： 沈星宇

学 号： 1951576

指 导 教 师： 张颖

学院、 专业： 软件学院 软件工程

同济大学

Tongji University

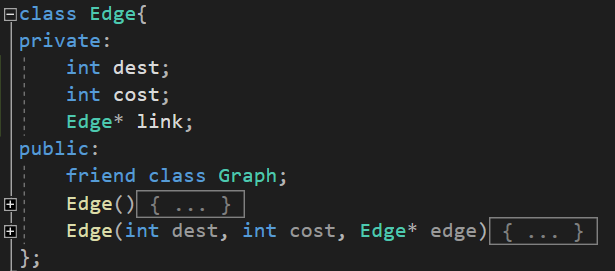
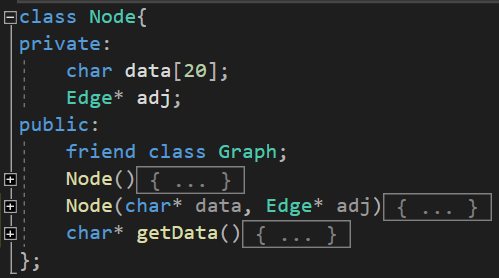
1. **分析**
2. **应用背景**

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。

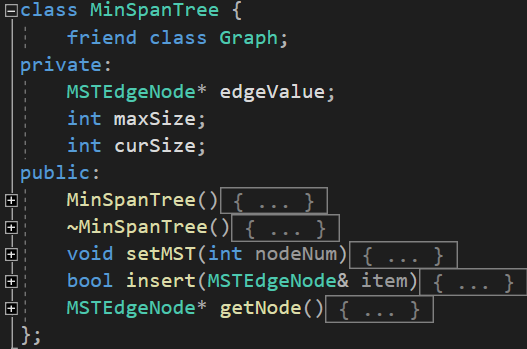
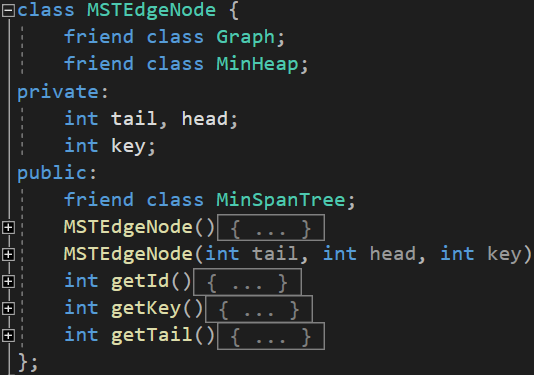
1. **项目功能要求**

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。

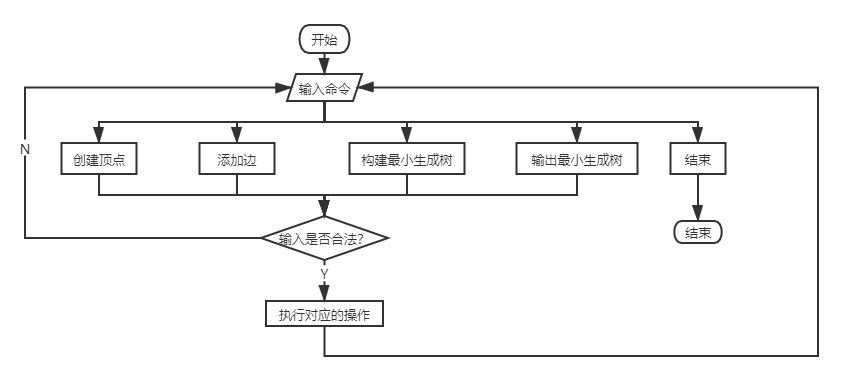
1. **设计**
2. **数据结构设计**

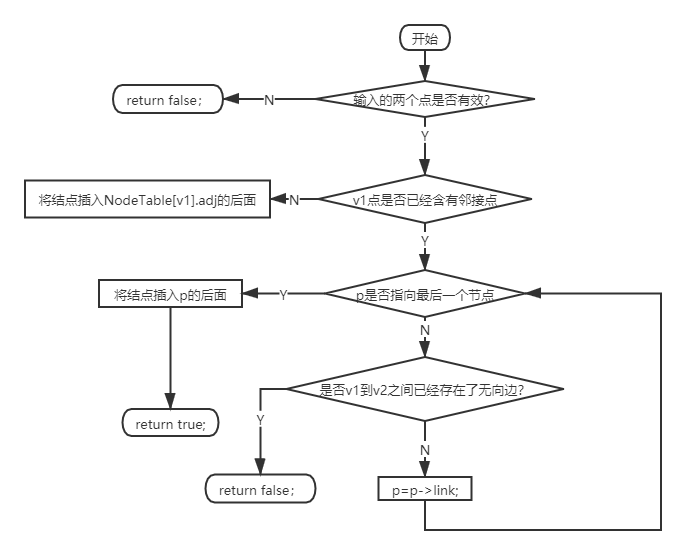


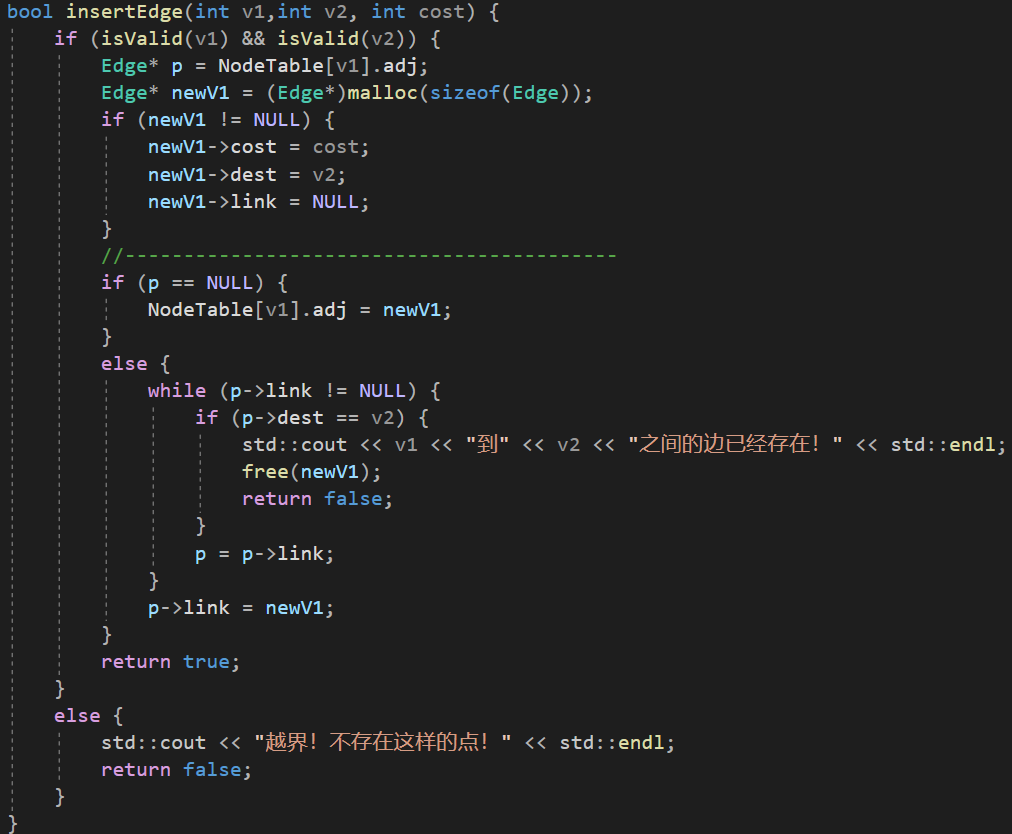
1. Graph是图，本项目是通过构建无向有权图来实现的，存储方式是邻接表，故Graph的成员变量有一个NodeTable，用来存储所有的结点。
2. Node是图的结点，内部的成员变量data用于存储结点的名称，Edge\* adj指向后面的Edge。
3. Edge是图的边结构，内部的成员变量dest是该边的下一结点序号（因为Node当中存储了第一个结点），cost是该边的权值大小，Edge\*link 链接了下一个边结构。

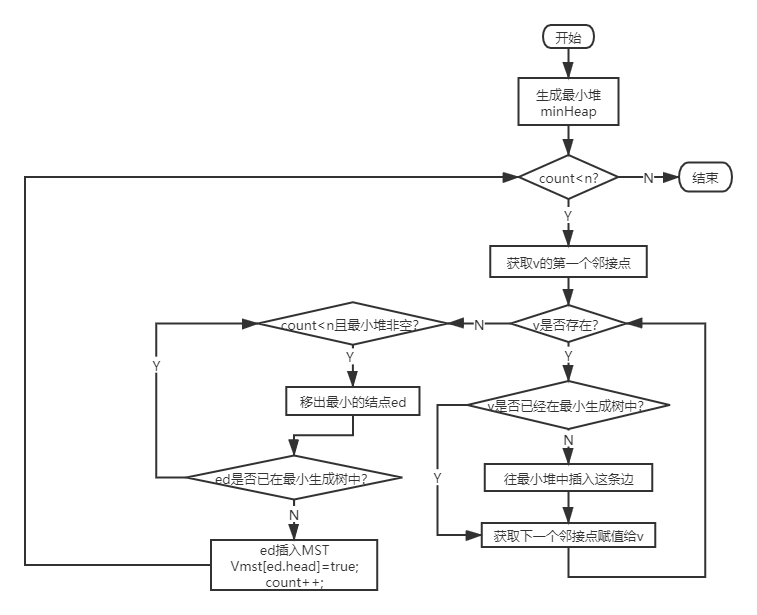


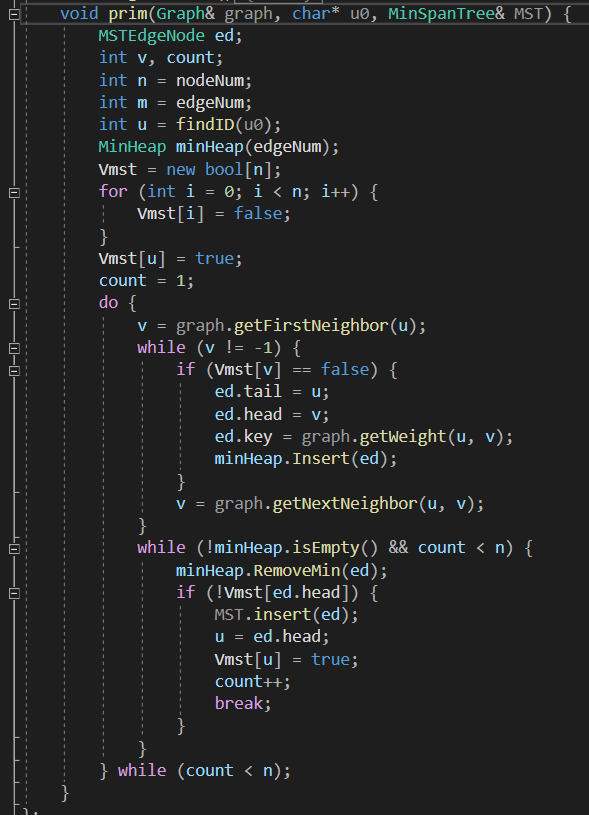
1. MinHeap是最小堆，本项目通过最小堆排序来取出最小值。其中函数void FilterDown(int start, int endOfHeap); 是用于自上而下将其调整为最小堆；void FilterUp(int i); 是将插入新节点后的结构从下往上调整为最小堆。函数bool Insert(const int& x);是往树里插入新的节点，而函数bool RemoveMin(int& x);是将根节点的最小值取出。
2. MinSpanTree是最小生成树结构，其实就是一个单链表，用于存储最小生成树的结果。
3. MSTEdgeNode是最小生成树的结点，head和tail分别代表该边的头尾结点序号，key代表边的权值。
4. **程序流程设计**
5. 首先程序会提示输入指令，会对指令的合法性进行判断。
6. 合法性判断包括：
7. 除A/a/B/b/C/c/D/d/E/e之外的所有指令均为非法，会被要求重新输入
8. 本项目的执行其实是有顺序的，比如以创建顶点、添加边、构建最小生成树、输出最小生成树的顺序进行，所以如果先执行了后续的操作命令，则会要求重新输入
9. 分别执行相应的操作



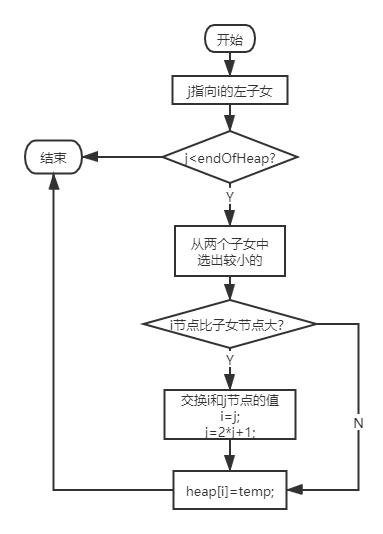
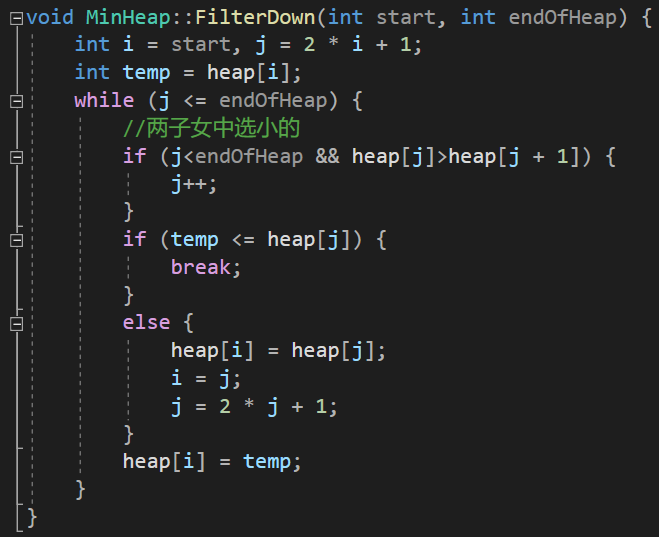
1. **实现**
2. **Graph::insertEdge()的实现**



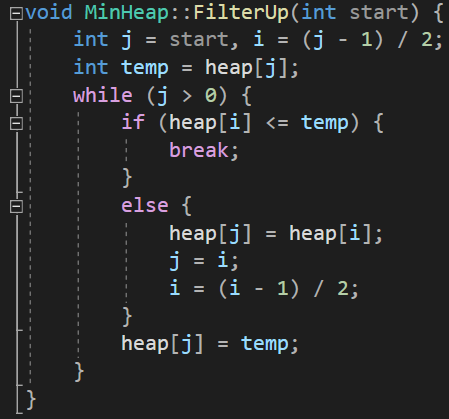
1. 首先会判定插入的边的端点是否存在，防止往不存在的顶点中进行插入
2. 然后判断v1对应的点是否已经有邻接点，如果没有则直接插入在NodeTable[v1].adj的后面；如果已有，则找该链的最后进行插入
3. **Graph::prim()的实现**



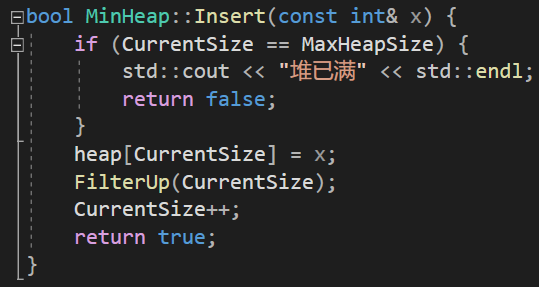
1. 图顶点集合为U，对任意选择的起始点u0，将该点加入集合V，再从集合U-V中找到另一点b使得点b到V中任意一点的权值最小，将b点也加入集合V；以此类推，直至所有顶点全部被加入V，此时就构建出了一棵MST。
2. 此处函数借用最小堆来寻找权值最小的边，用数组Vmst[]来记录是否将边放入了最小生成树。
3. **MinHeap::FilterDown()的实现**



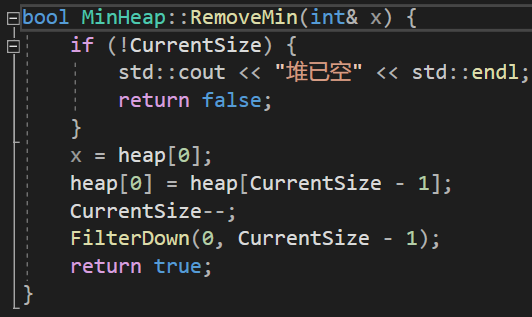
1. 于一个i来说，j=2\*i+1就是i这个节点的左子女（由堆的性质可得）
2. 当j到达堆的最后一个节点时结束循环
3. 从i的左右子女（如果由右子女的话）当中选出最小的那个，与i节点的值进行比较，如果i节点已经是最小的了，则结束循环；如果i的子女节点比i节点更小，那么交换两个节点的值，同时继续向下搜索（即i指向j，j指向下一个左子女）
4. **MinHeap::FilterUp()的实现**



1. 本函数是针对新加入的节点，从下往上调整它的位置
2. j指向当前节点，i是j的父母节点，循环判断新的节点的值与其父母节点的大小关系，如果新的节点比其父母节点更小，则往上调整其位置；其他的情况则可以退出循环。
3. 基本上就是MinHeap::FilterDown()的逆过程
4. **MinHeap::Insert()的实现**
   1. 将插入的值放在数组的最末端，然后调用FilterUp()函数将其调整为最小堆，同时调整最小堆的大小。



1. **MinHeap::RemoveMin()的实现**
   1. 移除根节点最小的那个数字，通过引用x来将其值传出
   2. 同时对于heap[]数组来说，要做的就是把最前面的数字给放到最后，同时让CurrentSize--,确保该值不会再被访问。
   3. 调用FilterDown()函数，调整其重新成为最小堆。



1. **测试**

1、**合法性检测**

（1）输入的指令不合法（即超出A/a/B/b/C/c/D/d/E/e）,会要求重新输入。

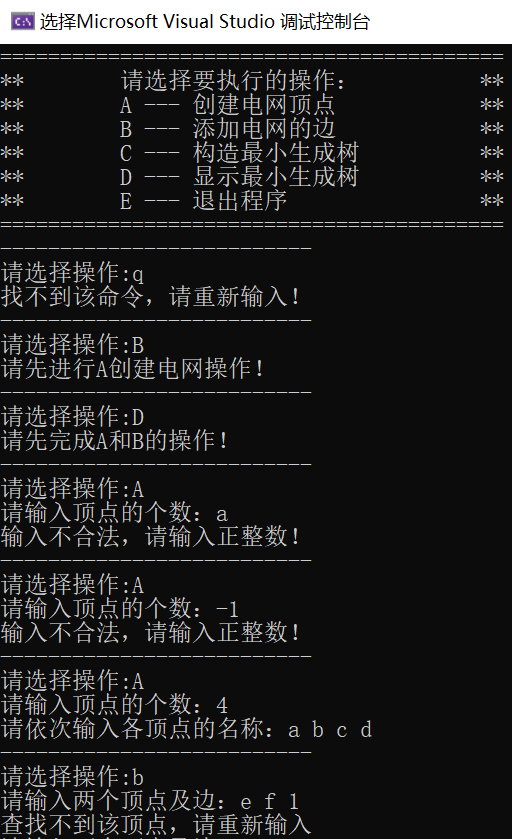
（2）本项目的执行其实是有顺序的，比如以创建顶点、添加边、构建最小生成

树、输出最小生成树的顺序进行，所以如果先执行了后续的操作命令，则

会要求重新输入

（3）如果输入的顶点个数/权值不为正整数，则会要求重新输入

（4）如果输入的点不再列表中，则会要求重新输入



**2、一般情况的功能实现**

