****

**Tongji University**

**离散数学实验报告**

**004-最小生成树**

专 业： 软 件 工 程

指导教师： 唐 剑 锋

学 号： 2 1 5 3 0 6 1

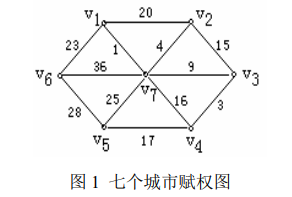
姓 名： 谢 嘉 麒

1. **实验概览**
   1. **实验背景**

城市通信网路的建造是关键的民生问题之一。通讯网建设需要将保证信号能够传输到所有村庄聚落，而在此基础之上，需要考虑通讯网的建设造价问题。整个通讯网络可以看做一张图，而求通讯网络最低造价即是求加权边长的图的权最小构造方法，即最小生成树。

* 1. **实验要求**
     1. **实验整体要求**

如下图所示的赋权图表示某七个城市，预先计算出它们之间的一些直接通信道路造价（单位：万元），试给出一个设计方案，使得各城市之间既能够保持通信，又使得总造价最小，并计算其最小值。

****

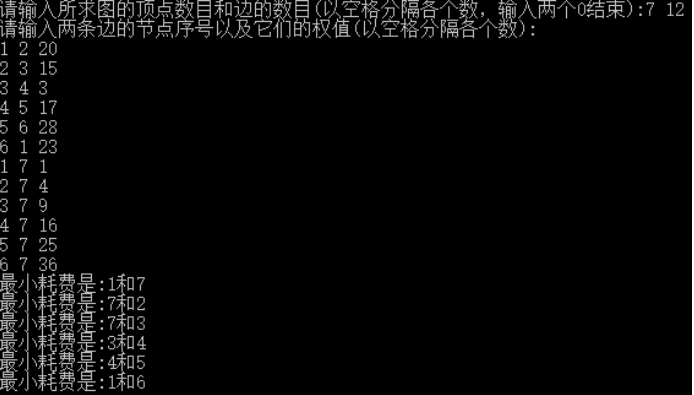
* + 1. **实验输入要求**

项目输入要求实现客户手动键入顶点数和边数，程序得到之后提示用户键入各定点及其间的加权边。

* + 1. **实验输出要求**

要求项目输出最小通讯网络的各个边两头的结点。

* + 1. **实验示例**

****

* 1. **实验思路分析**

根据项目要求可以得到，本次项目的实质是将通讯网的各个城市点视作图的各个顶点，将通讯线路视作图的边。求通讯道路造价最小即是求最小生成树的过程。因此本项目需要实现图的数据结构，并在此基础上利用Kruskal算法或者Prim算法实现求最小生成树。

* + 1. **实验功能完善**

由于通讯网络是关键的民生问题之一，需要程序功能完善，保证用户能够创建通讯网络顶点，创建网络各个通讯线路，生成最小生成树以及打印最小生成树的几条边。在用户输入时要进行输入错误处理并给出合理提示。

* + 1. **执行效率优化**

为了提高用户的使用体验，需要优化程序的执行效率。这一点可以从程序的空间开销和时间开销入手，尽量避免不必要的递归、循环等。

* + 1. **代码的健壮性**

由于程序在用户生活中起到重要作用，故需要保证代码的健壮性。当用户进行不正确使用时，代码需要进行报错提示，避免崩溃退出。同时也要保证存储结构封装安全，数据存储严谨无遗漏。

1. **数据结构的设计与使用**
   1. **图类的设计**

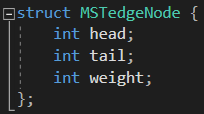
图（graph）是一种灵活的非线性结构。图一般由顶点集合（vertex）及顶点间的关系集合组成的一众数据结构：Graph = ( V , E )。其中顶点集合是有穷非空集合，边（edge）集合是顶点间关系的有穷集合。

由于本次项目中通讯网络各个城市站点之间是由无向输通讯线相连，因此可以设计成为无向图。

对于图存储结构的选用，包括邻接矩阵法和邻接表法。考虑到两种存储方法在执行遍历操作和Kruskal/Prim算法时，邻接矩阵具有简洁便利的特点，故本次项目选择使用邻接矩阵存储图。

1. 定义最小生成树结点结构体：

在结构体内定义边的头、尾和边权值等变量，均为整形数据类型。将此结构体作为存储最小生成树各边的基本单元参与程序。



1. 利用邻接矩阵存储图的信息：

邻接矩阵存储法即假设图有n个结点，设定n\*n的矩阵，将矩阵内容初始化为INT\_MAX，若顶点<i,j>间存在有向边，则将矩阵i行j列元素置为其边的权值。无向图需要同时将j行i列也置为其边的权值。



* 1. **并查集的设计**

实现最小生成树Kruskal算法需要使用并查集作为底层数据结构辅助算法。

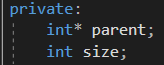
建立等价类的一种解决方案是先把每一个对象看作是一个单元素集合，然后按一定顺序将属于同一等价类的元素所在的集合合并。在此过程中将反复地使用一个搜索运算，确定一个元素在哪一个集合中。能够完成这种功能的集合就是并查集。

它支持以下三种操作：union (Root1, Root2) //并操作；find (x) //搜索操作；UFsets (s) //构造函数。

本次实验将并查集作为一种单独的类进行封装和实现，具体如下：

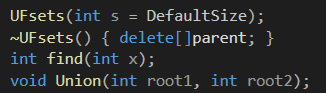
1. 私有化封装并查集变量：

设定int\*型父母指针和int型变量size。

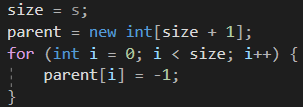


1. 公有化调用类成员函数：

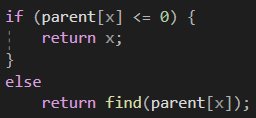
设定类构造函数，析构函数和并操作，搜索操作；



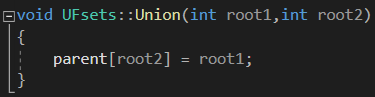
1. 在并查集类的构造函数中为父母指针数组动态分配大小为size+1的内存空间。在析构函数中用delete语句释放内存。



1. 在find函数中使用递归方法搜索并查集



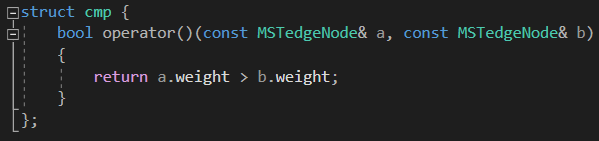
1. 在union函数中将并查集合并



* 1. **最小堆数据结构的使用**

最小堆作为优先级队列中最高效的一种数据结构，有着很广泛的应用。其主要思想是将较小元素上浮至堆顶（即队列队首）

本次实验调用STL库queue和vector作为最小堆的前置条件。为了方便最小生成树结点的存取和权值比较，同时定义了cmp作为最小堆的比较方法，具体如下：



利用这一比较方法，则可以定义最小堆：



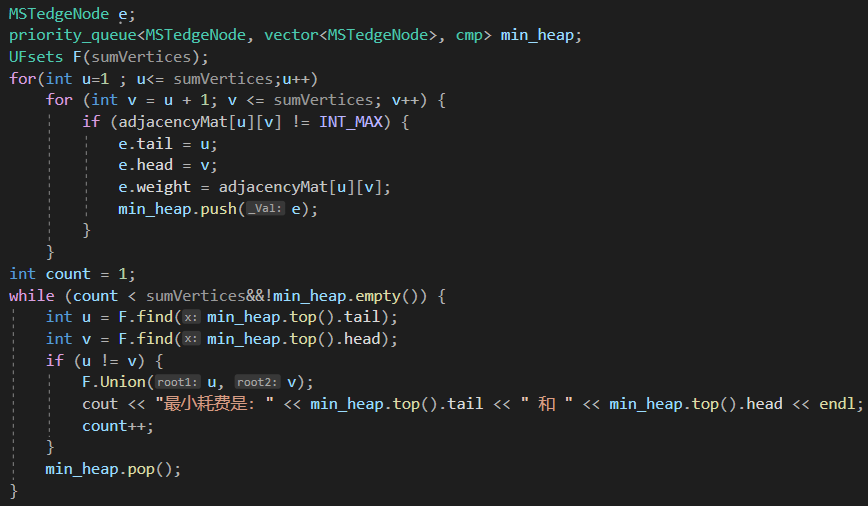
最小堆的插入，取顶元素，删除只需要调用系统函数.push()、.top()和.pop()即可。

1. **算法的设计与使用**
   1. **Kruskal算法的设计与使用**

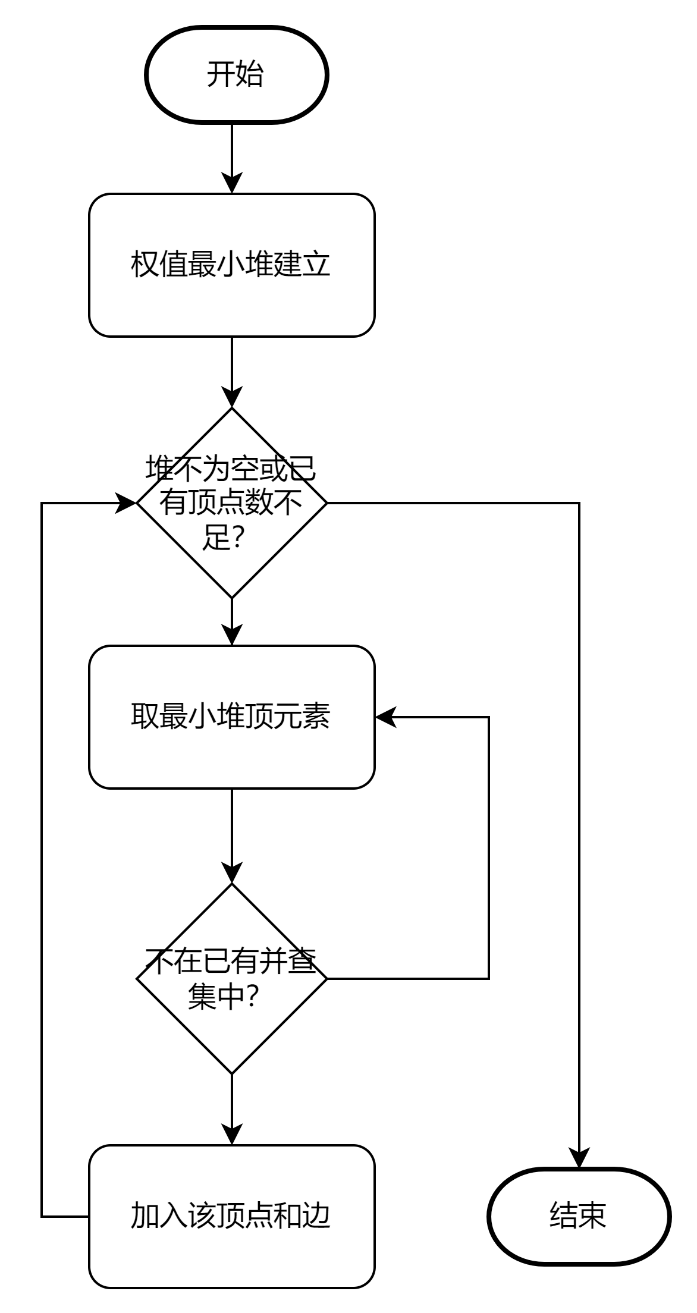
为了求解最小代价，使花费的总代价最小，这是数学中经典的求解最小(费用生成树的算法。其构造可以采用Kruskal算法或Prim算法进行实现。

Kruskal 算法的核心思想是：

1. 假设该图 G 是不连通的，对该图的边以权值非降序重新排列；即将边及其首位结点和权值存储在最小堆中，最小堆会自动将边按照权值从小到大的顺序进行排列。
2. 对于排序表中的每条边，如果现在把它放入最小生成树 T 不会形成回路，则把它加入到 T 中；否则丢弃；是否形成回路需要看新点是否存在于已有生成树的连通分量中。这一点可以通过并查集实现。求不在同一连通分量中的点即是求不在同一并查集中的顶点。
3. 输出最小生成树 T 的结果，得到想要的答案。
   * 1. **Kruskal算法代码实现**

****

* + 1. **Kruskal算法流程图**



是

否

是

否

* 1. **Prim算法的设计和使用**

连通图中的每一棵生成树，都是原图的一个极大无环子图。因此从中删去任何一条边，生成树就不再连通；反之，在其中引入任何一条新边，都会形成（恰好）一个回路。

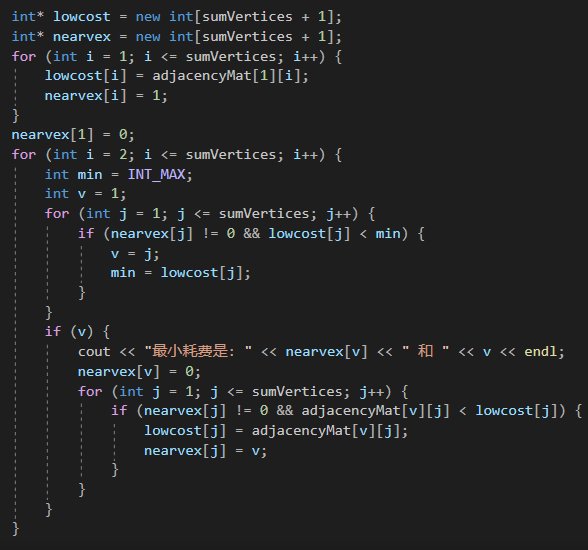
若连通网络由n个顶点组成，则其生成树必含n个顶点，n-1条边。因此，构造最小生成数的准则有三条：

1. 只能使用该网络中的边来构造最小生成树；
2. 只能使用恰好n-1条边来联络网络中的n个顶点；
3. 选用的n-1条边不能构成回路。

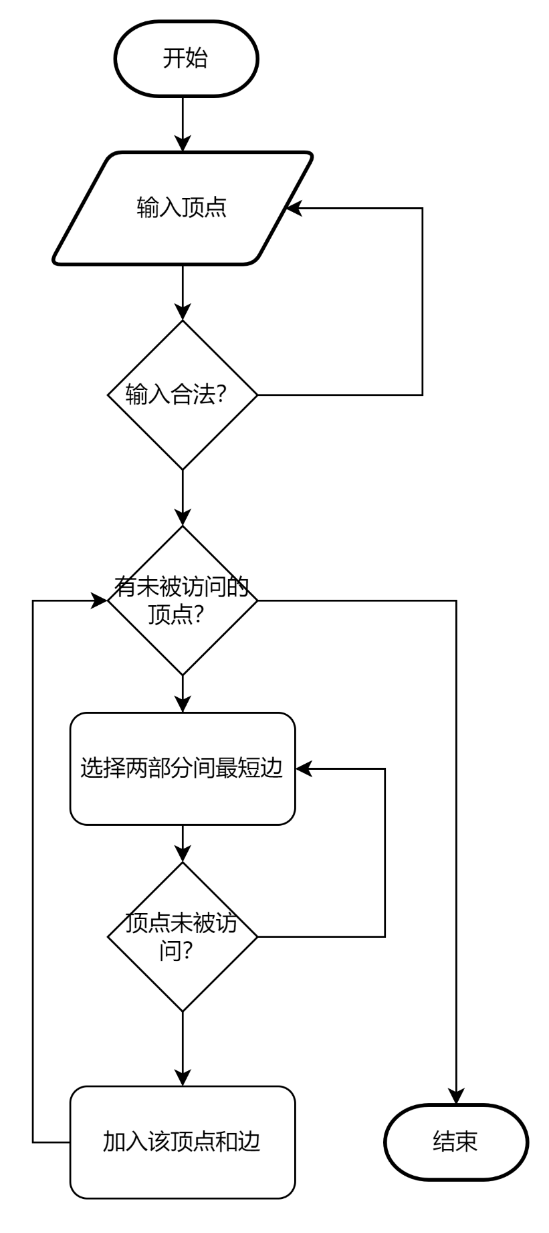
Prim算法采用了逐步求解（grady）的策略，是一种贪心策略。

算法主要思想是迭代，每次迭代都将顶点分成在最小生成树中和不在最小生成树中两部分，选择两部分间权值最小的边作为最小生成树的新边，随其顶点一起加入最小生成树。因此每一轮迭代都会使最小生成树中新增加一点一边。当其包含n个顶点，n-1条边时，算法即可终止。

* + 1. **Prim算法代码实现**

****

* + 1. **Prim算法流程图解**



否

是

否

是

是

否

1. **心得体会**
   1. **对于基础知识的巩固和理解**

本次实验巩固了对于图类型的理解和学习，尝试通过构造最小生成树求最短路径和最短造价问题。将已学到的离散数学知识和生活实际应用联系起来，通过实践检验知识的正确性，也通过知识来指导实践，做到知行合一。

* 1. **对于算法的学习和设计**

通过本次实验接触并学习了贪心算法的概念，设计并使用了其中的Kruskal算法和Prim算法。

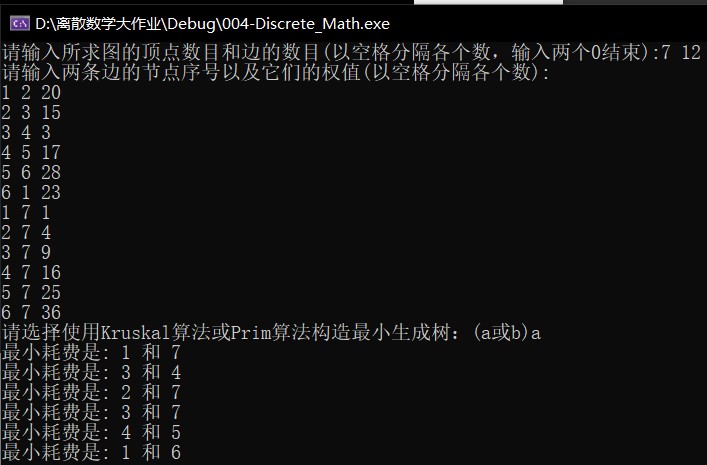
在设计算法的过程中，一方面广泛搜集新知识，通过查找等方式学习了并查集，最小堆等数据结构的设计原理和使用方法，掌握了STL中queue队列的使用，并利用优先级队列构造最小堆；

另一方面则通过算法的具体设计发现了两种算法各自使用场景的优劣：Kruskal算法是一种对点的贪心算法，更适用于点稀疏而边稠密的情况；而Prim算法则是一种对于边的贪心算法，更加适用于边稀疏而点稠密的情况。

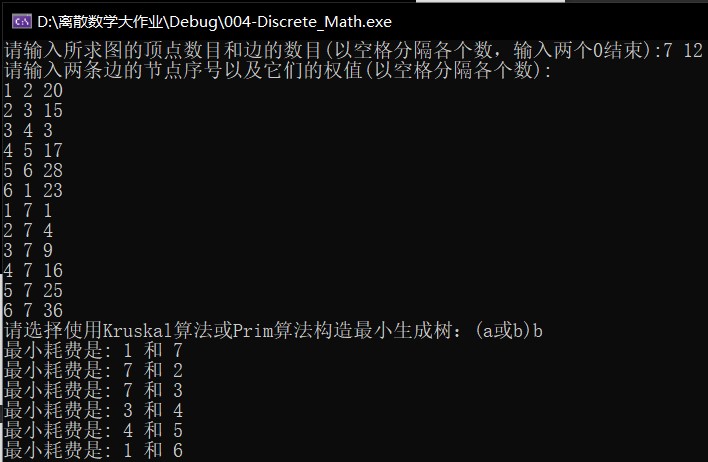
* 1. **设计实验的能力提高**

通过不断地实验设计，更加理解并优化了实验的解决过程。通过设计实验思路，设计数据结构，优化算法等操作逐步解决实验，提高了解决问题的能力和编程、算法能力。

1. **程序测试**
   1. **Kruskal算法生成最小树测试**

****

* 1. **Prim算法生成最小树测试**

****