**计算机科学与技术学院课程设计成绩单**

课程名称：数据结构课程设计

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 | 卢俊朴 | 性别 | 男 | 学号 | | 202113407120 | | 班级 | 软件工程2101 |
| 电话 | 15527752855 | | 综合成绩 | |  | | 成绩等级 | |  |
| 程序运行情况  （占总成绩20%） | |  | | | | | | | |
| 程序功能完善程度  （占总成绩20%） | |  | | | | | | | |
| 对问题的答辩情况  （占总成绩40%） | |  | | | | | | | |
| 学生的工作态度与独立工作能力  （占总成绩10%） | |  | | | | | | | |
| 设计报告的规范性  （占总成绩10%） | |  | | | | | | | |

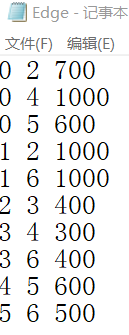
A：90~100分 A-：85~89分 B+：82~84分 B：78~81分 B-：75~77分

C+：72~74分 C：68~71分 C-：64~67分 D：60~63分 F：<60分

武汉科技大学计算机科学与技术学院制表

**【题目】旅游区景点导游系统**

1. **数据格式**



用了Edge和Vex两个文本文件（.txt类型 txt文件编码类型为ANSI不是此类型读取文件会乱码）

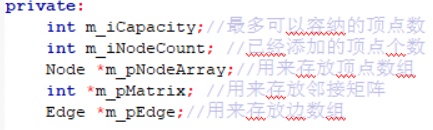
Vex文件储存了点的：初始点个数（第一行int），然后每三行依次是：景点序号（int）；景点名称（string）；景点介绍（string）。

Edge文件储存了边的：起始景点，结束景点，权重（每行都是这三个，都是int）。

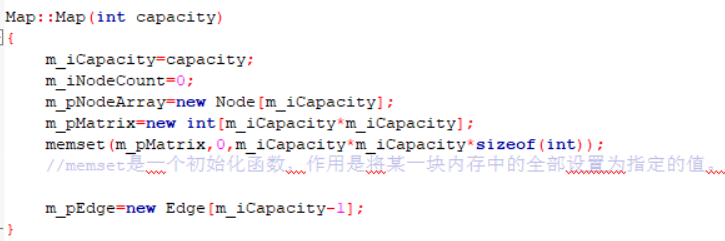
1. **数据结构（读文件创建图）**

用#include<fstream> 中的 ifstream将文件中的数据读入内存，用的是邻接矩阵作为存储结构。

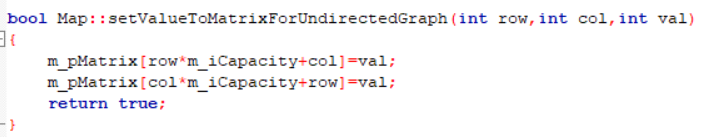
先构造定义Edge类和Node类(构造见1数据格式)，在图类的private属性中用Edge类和Node类的指针分别存放顶点和边的数组，再用一个int指针存放邻接矩阵。如下：



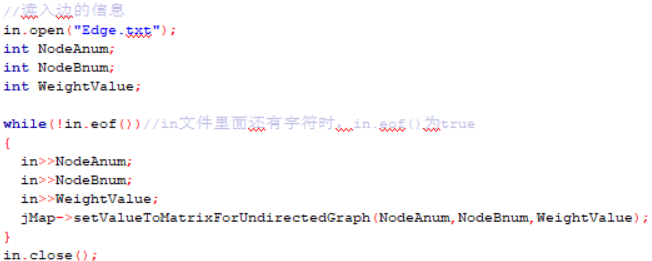
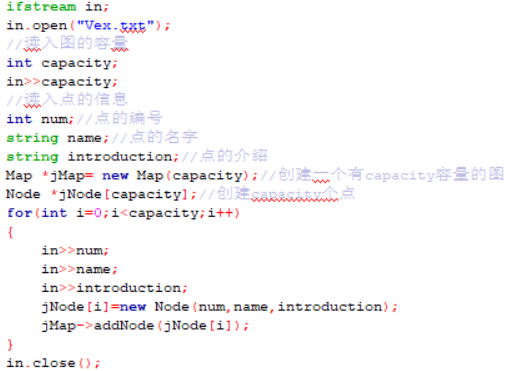
图的构造函数如下：



读入Edge.txt内容创造邻接矩阵 函数如下：

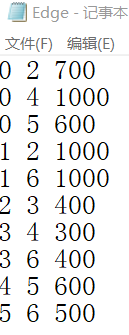


文件读入内存函数：



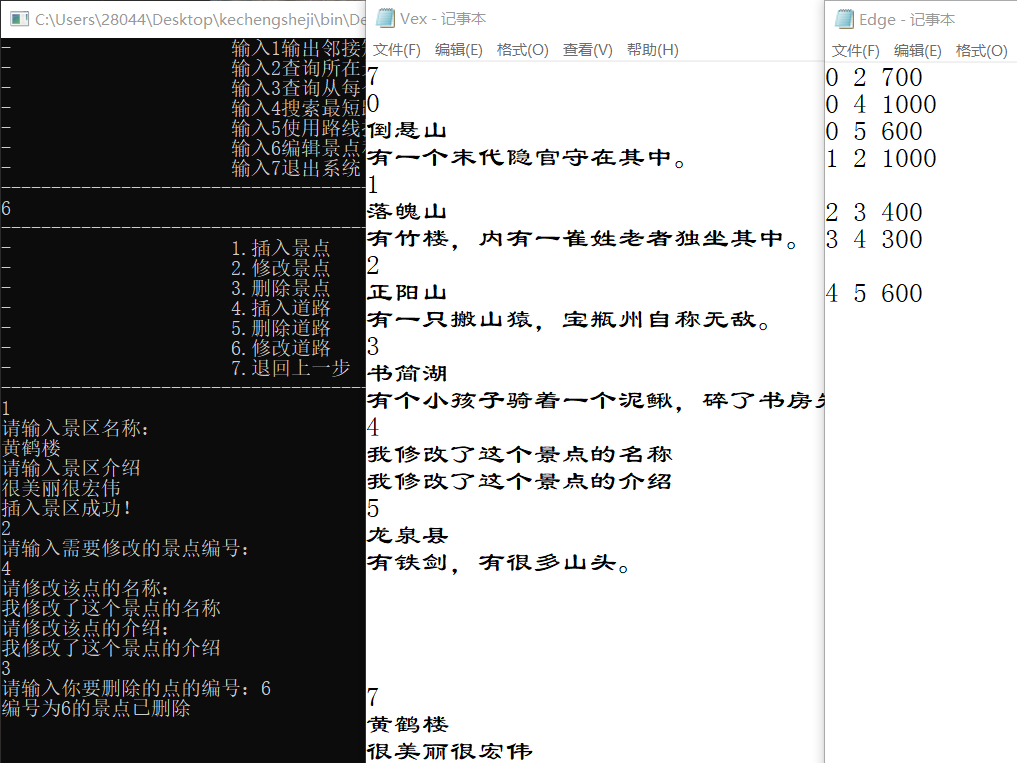
1. **查询、编辑景点信息（为下面的步骤均用原始的景点和边演示）**

**原始Vex和Edge：**



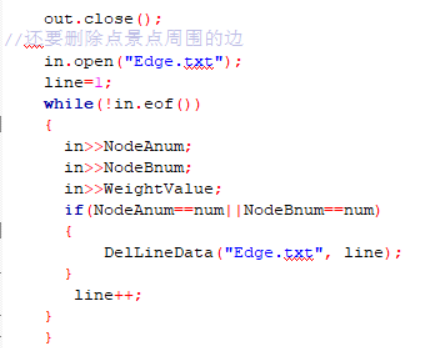
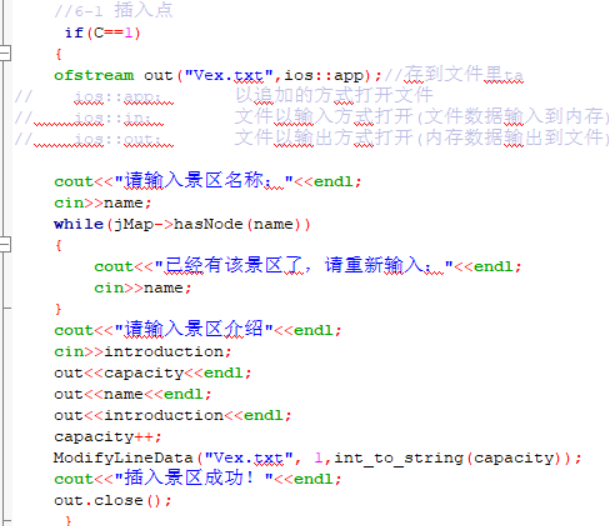
添加，删除，修改景点后运行结果以及编辑后信息保存回文件：

依此插入，删除，修改景点后vex与edge记事本的变化为如下图

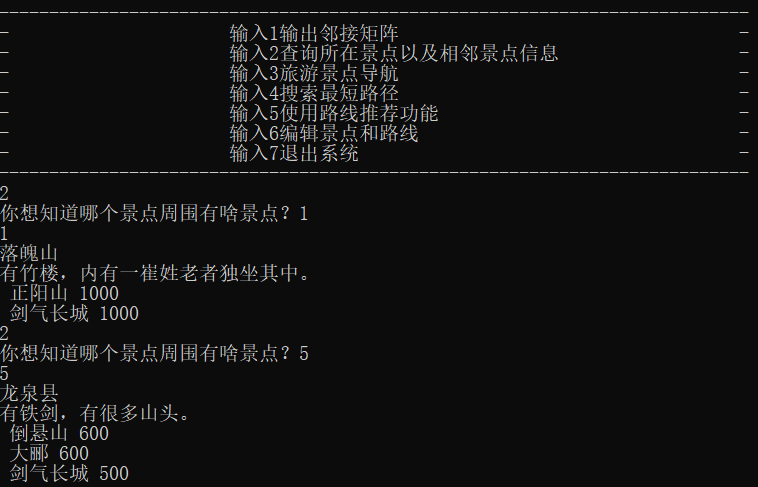


补充说明：增加了黄鹤楼和其介绍，修改了4的内容，删除了景点6。

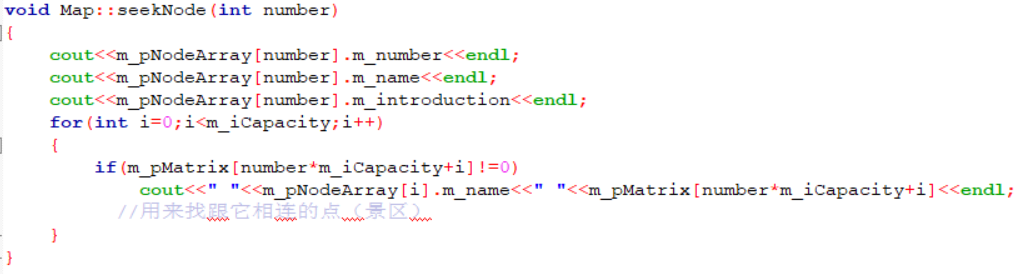
具体代码为：



1. **旅游区景点显示**

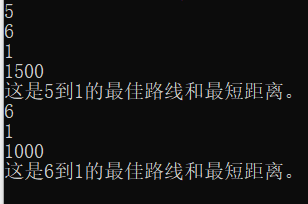
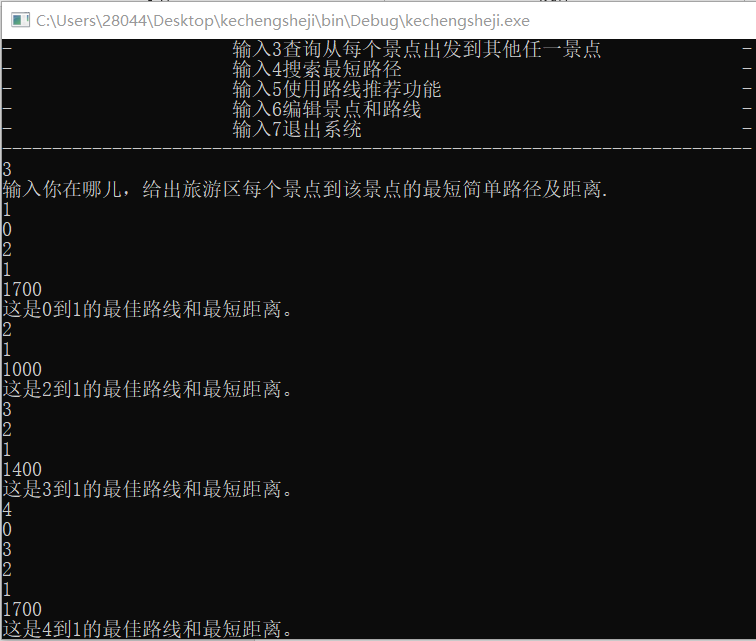


**主要代码为：**

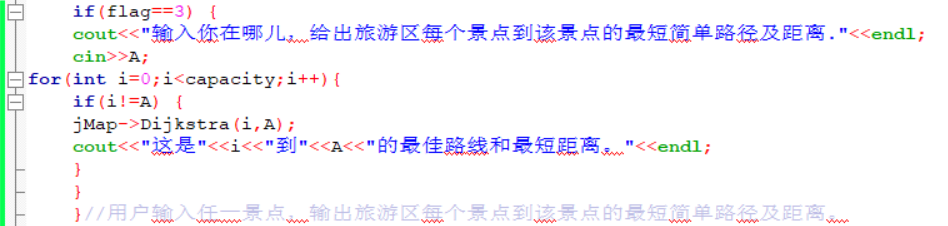


1. **查询从每个景点出发到其他任一景点的最短简单路径及距离**

运行结果：



算法：迪斯科算法

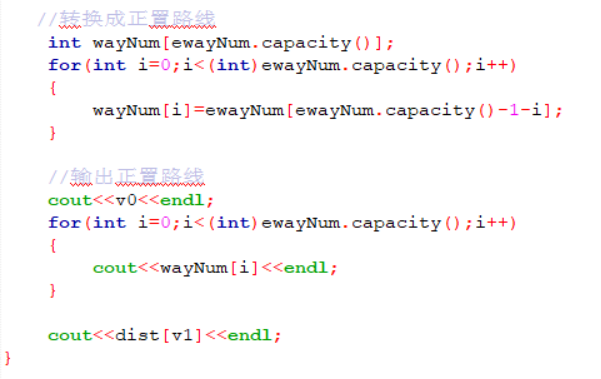
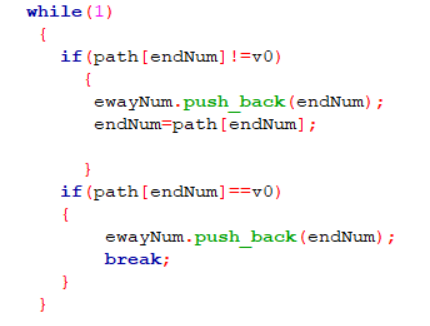
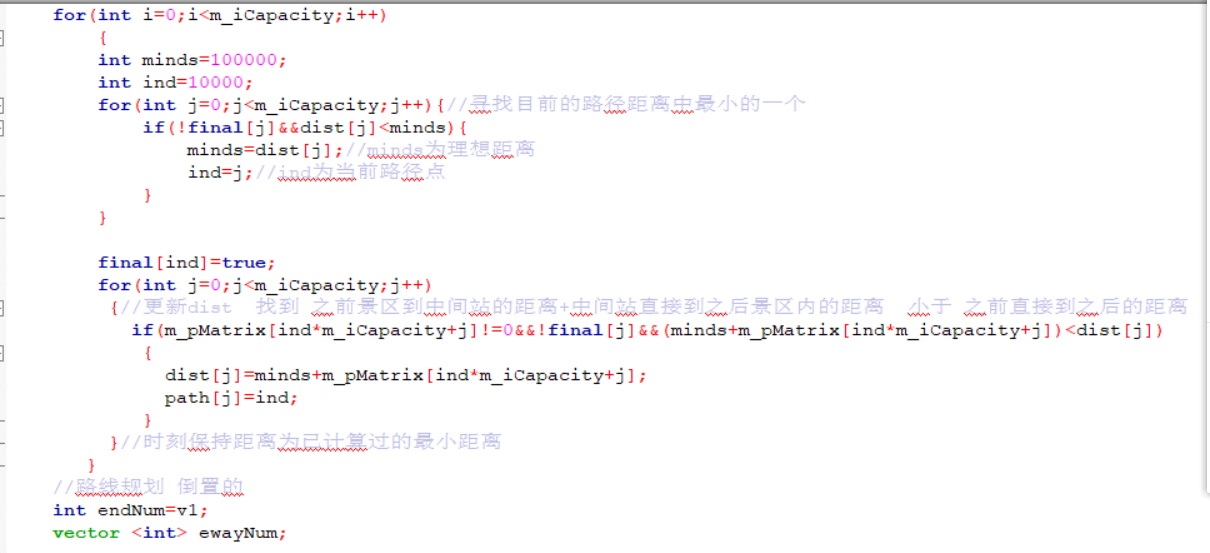
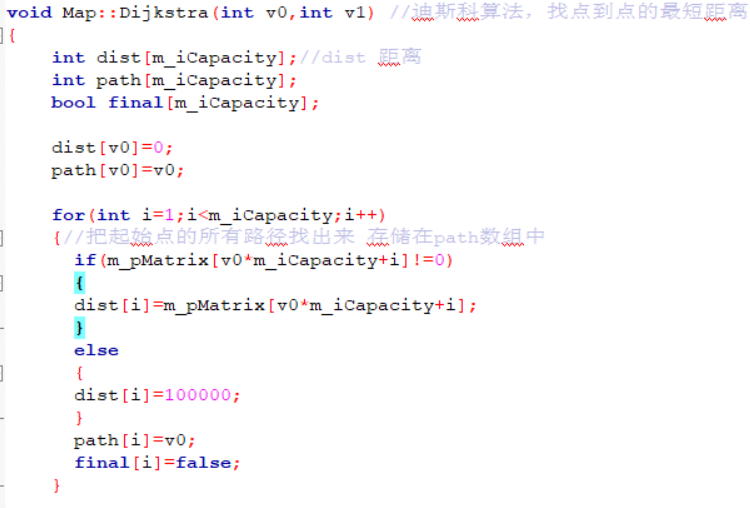


Dijkstra(i0,i1) 输出i0到i1之间的最简单路径 以及 距离

使用循环实现所有非i1点到i1的最佳路线以及最短距离。

Dijkstra详细描述以及代码如下：

图一→ 图二↓



图三↑ 图四↑

补充解释：dist为总权值的数组，path为路径数组，代表从dist[v0]，及vo到其他边的总权值(不断更新直到最小)

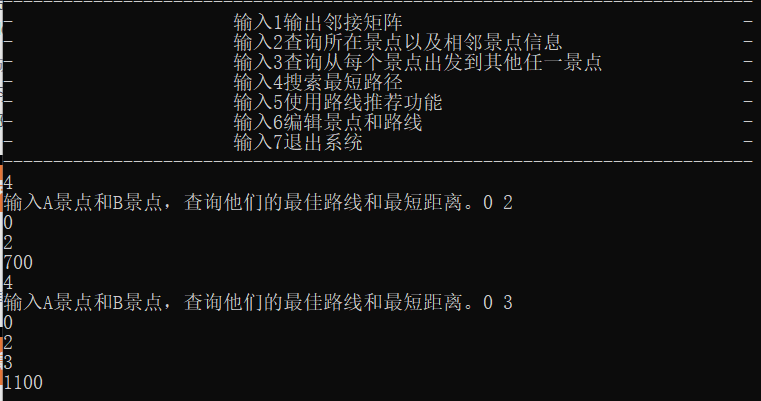
图一为用邻接矩阵的权值不为0 找到v0 景点的所有相邻路径 储存在path数组中，不直接相邻则设置为无穷大(100000)，final为没有达到最短距离。

图二为寻找当前在数组中的元素中dist最小的一个，把他确定下来(之后不管他)，然后从确定的path点开始继续向后找，这个点到其他点的距离。如果这个点和之前的点都到了另一个点path[j]，则比较他们的大小，然后选择最小的。不断重复，直到v0到所有的点的最短距离全部确定。

图三为创建一个容器vector<int>ewayNum，倒置的依此从v1开始倒置存放经过的路线在ewayNum容器中(因为用的是类栈的pushback所以是倒置)，直到v0==v1(及从v0景点到v1景点)

图四为把ewayNum容器正放在wayNum数组中，然后输出正置路线(一个一个输出所有的经过)和总路程。

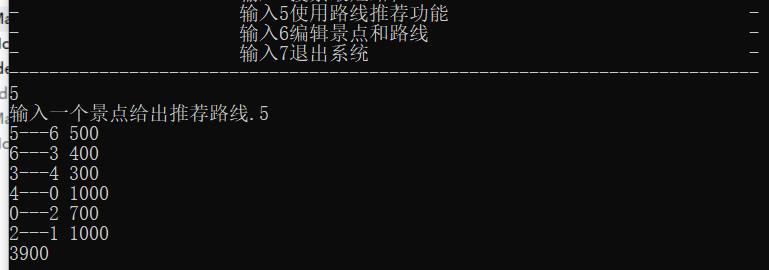
**6、查询任意两个景点之间所有简单路径及距离、最短简单路径及距离**



Dijkstra详细实现与 6,查询任意两景点之间的所有简单路径及距离相同，详细见5的Dijkstra详细描述以及补充解释。直接用Dijkstra函数，比5简易

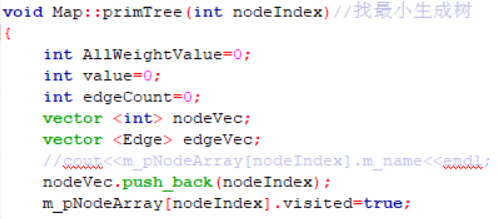
**7， 最佳游览路线推荐**

**程序运行结果如下：**



**运用primTree算法，找最小生成树**

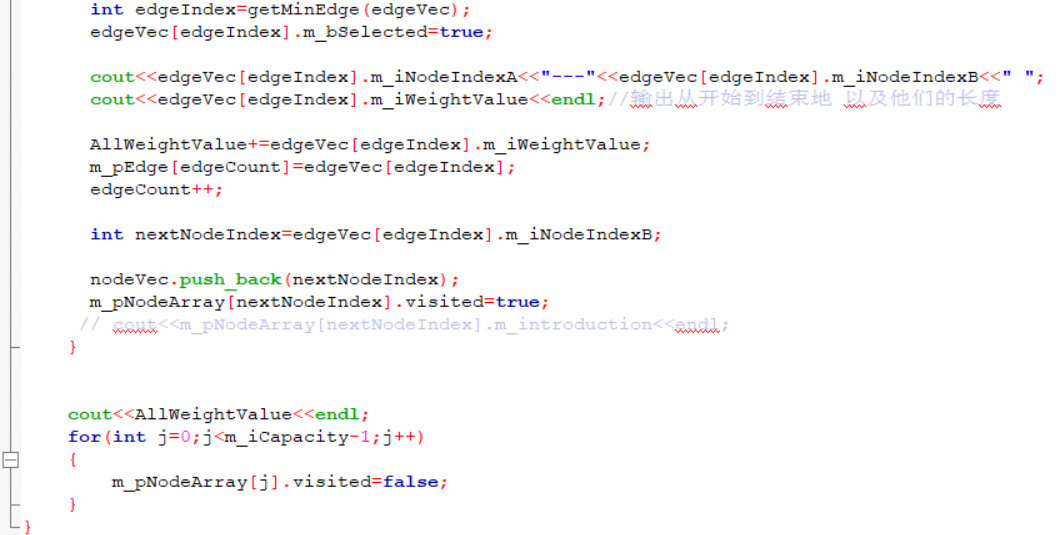
**primTree详细代码以及描述如下：**



图一↑ 图二↓



图三↓



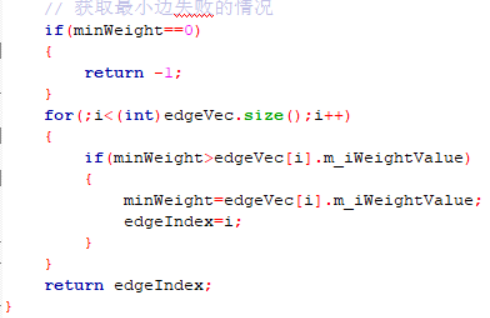
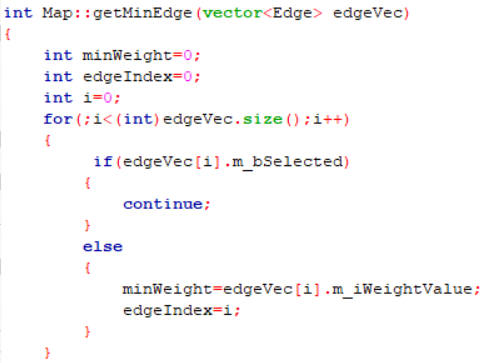
补充解释：

图一：创建两个容器，一个点容器一个边容器，把起始点放进点容器，并设置为已访问

图二：从点容器的第一个元素开始(第一次循环为起始点)，把所有相邻的点和边分别放入点容器和边容器中。

图三：从点容器中选出最小的边，然后设置为已选择，输出边容器的内容(起始点，结束点，距离)，把距离加入总距离中，然后把结束点存放点容器，让下一期循环的起始点是它。然后重复图二图三，直到最小树的景点个数与总景点个数一样。最后再把图内的点数组设置成未访问，以便后续操作。

图三的getMinEdge函数如下：



**8、设计总结**

初见作业：

一开始看到这个作业我是懵圈的，我根本不知道是用啥写，我想了上学期学的java，.net，数据结构，我发现，哪一个我都不会写，因为数据结构的学习中，我根本不知道怎么把学的东西进行实际应用，更不知道怎么把创建的图和学到的各种结构联系起来，直到我翻看了无数CSDN的博客，看了各种各样的课程设计，我才能逐渐的完善这个课程设计的作业内容，用了非常久的时间。后来兴致冲冲的打印出来了发现还是有问题，后又重新更改打印。

遇到的问题：

包括但不限于：忘记codeblocks怎么使用；怎么创建图以至于更好的和函数联系在一起；怎么读文件里的数据怎么更改文件里的数据；各种函数(普利姆，迪斯科……)怎么实现；为什么我的图读出来中文乱码；为什么我的图创建后编译器直接闪退；为什么我的函数写出来运行不了；为什么我运行不了以前老师发的文件；为什么老师发的构图步骤和网上的差别那么大，我应该用哪个，更改景点和边时怎么让它们互不影响又互相影响(没有完全解决)，用算法时怎么才能契合自己创建的图等等。

解决过程：

CSDN上搜，阿里云上搜，github上搜，CSDN上充钱买资源学习。然后结合老师发的关于图的各种操作，增删改查，拼凑重构。

体会：

感触很大，一开始自己照着老师发的图的代码写程序，不是这里出错就是哪里出错，后来在网上找了很多类似的图的题目，才找到了自己能理解的方法，逐渐完善了课程设计。明白为什么学计算机的大多是秃头了，计算机之路任重而道远，以后要主动学习，不能只拘泥于老师讲的，而且研究代码看不懂时不要死看，要带进去算一算，不然很伤脑经。