**校园导游咨询**

1. **实验内容**

【问题描述】

1. 设计你学校的校园导游图，所含景点不少于十个，以图中顶点表示各景点，存放景点名称，代号，简介等信息，以边表示路径，存放路径长度等相关信息。
2. 为来访客人提供图中任意景点的问路查询，即查询任意两个景点之间的最短简单路径。
3. 为来访客人提供图中任意景点相关信息的查询。

【测试数据】

根据济南大学实际情况自定义一张校园平面图

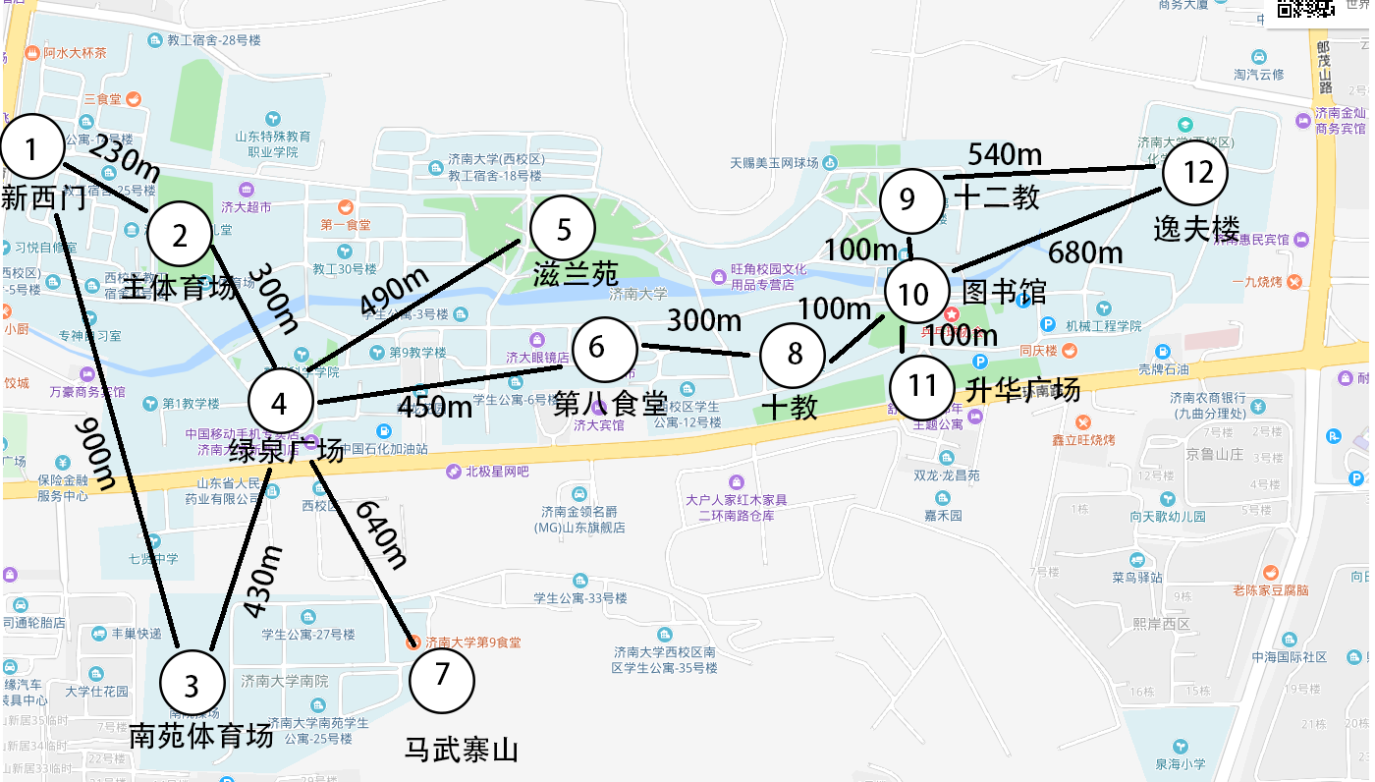
【实现提示】

一般情况下，校园的道路是双向通行的，可设校园平面图是无向图，用dijksrta算法计算最短路径，参考教材P376-379

【选做内容】

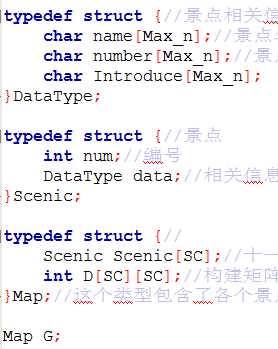
1. 校园导游图可以根据需要随时增加景点和道路
2. 可以求出任意两个景点之间的所有路径
3. 多个景点间的最佳访问路线查询，即经过这多个景点的最短路径。
4. **数据结构设计**

地图：



采用邻接矩阵的方法存储图以便于计算。

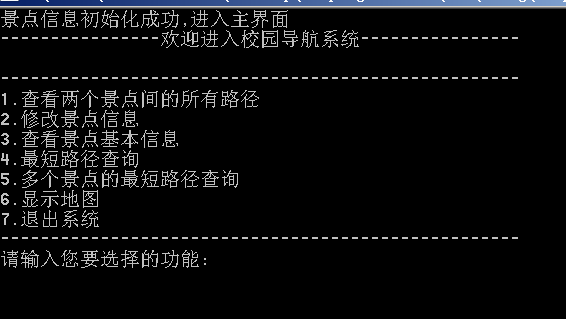
数据结构如下



DataType用于保存景点的信息，Scenic用于保存景点序号，Map生成地图，保存景点和地图，其中地图用二维邻接表保存。

**三、算法设计**

主界面：



（1）对于功能1，采用递归深度优先遍历的方法获取以要求节点为起点的所有路径。

时间复杂度为O(n^2).

（2）对于功能2（增加景点或道路）直接对邻接矩阵进行修改，增加一个景点时即为增加一行一列，默认所有元素为无穷大，然后将本元素与本元素相连位置的值置为零。

增加一个路径时，将地图邻接矩阵相应路径及相反路径值为相应的距离。

（3）对于功能三，直接根据下标获取要查询的节点的信息并输出。

（4）对于功能4，采用 dijksrta算法获取单个原点到其他点的最短路径（时间复杂度为O（n^2）），用目的节点获取最短路径，prev数组保存路径中前一个节点的信息，通过prev数组获取路径中经过的景点的序号，通过下标获取名称。

（5）对于功能5，采用分治法对输入的景点进行全排列，将全排列后的数组中任意相邻两个节点间的最短路径用dijkstra算法求出，将所有相邻接点的距离求和，对N种全排解求得N个和，最小值即为经过这多个路径的最短距离。最短路径为输出最短路径对应的全排解数组，带入景点中得到名称。

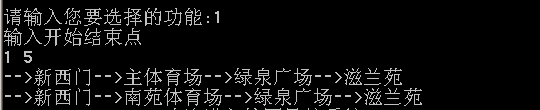
（6）对于功能6，直接通过系统命令 mspaint map.png 显示图片。

**四、测试数据及程序运行情况**

功能1：

测试数据：1 5

运行结果：

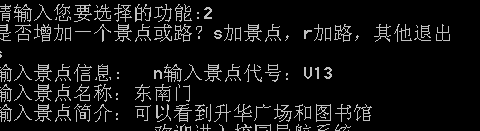


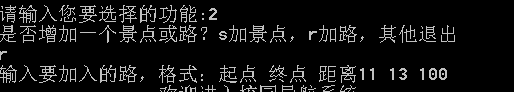
功能2：

测试数据：

（1）s V13 东南门 可以看到升华广场和图书馆.

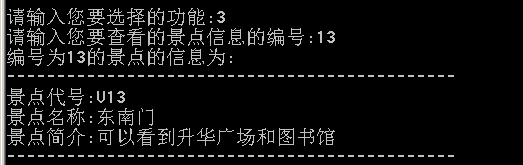
（2）r 11 13 100



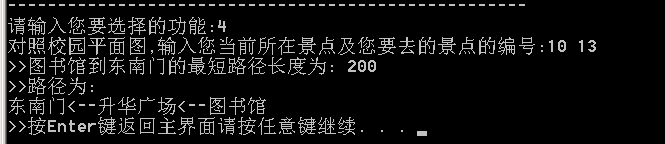


程序执行结果：

景点查询**：**



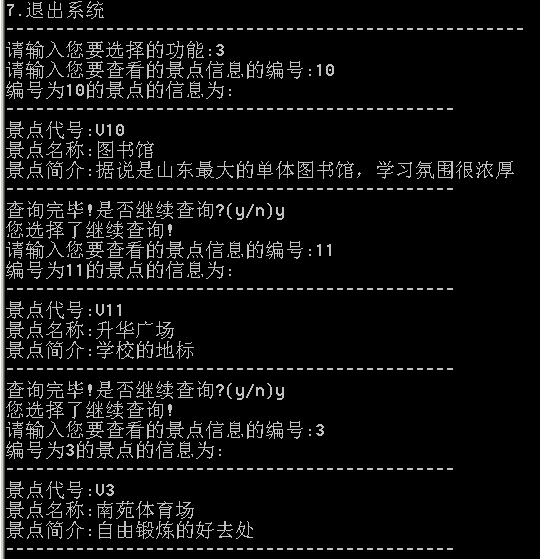
最短路线查询：



功能3：

测试数据： 10 11 3

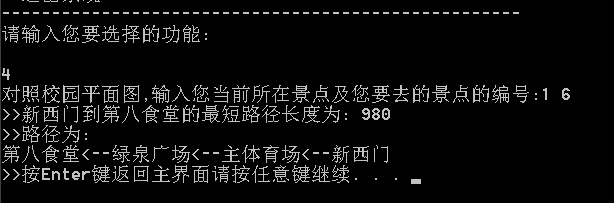
程序运行结果：



功能4：

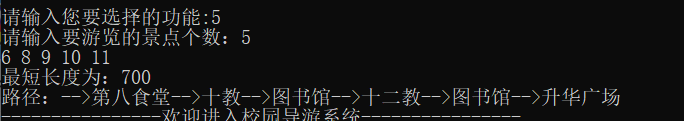
测试数据 1 6

程序运行结果：



功能5：

测试数据：5 6 8 9 10 11



通过长度为700可以看到选择对了最短的路径，重复经过的景点显示了两次。

**五、实验过程中出现的问题及解决方法**

对于功能5，一开始没有实现将dijkstra寻找到的最短路径以文字输出，因为输出全部路径时出现错误，仔细研究后发现是因为防止重复而少输出的一个节点在只有两个时会确实，后来修改了输出逻辑，加上可能重复的点，在输出时再忽略他，就实现了本功能。

**六、自我评析与总结**

通过此次课程设计，使我更加扎实的掌握了图论的知识，在设计过程中虽然遇到了一些问题，但经过一次又一次的思考，一遍又一遍的检查终于找出了原因所在，也暴露出了前期我在这方面的知识欠缺和经验不足。实践出真知，通过亲自动手制作，使我们掌握的知识不再是纸上谈兵。在课程设计过程中，我们不断发现错误，不断改正，不断领悟，不断获取。最终的检测调试环节，本身就是在践行“过而能改，善莫大焉”的知行观。这次课程设计终于顺利完成了，在设计中遇到了很多问题，最后在老师的指导下，终于游逆而解。在今后社会的发展和学习实践过程中，一定要不懈努力，不能遇到问题就想到要退缩，一定要不厌其烦的发现问题所在，然后一一进行解决，只有这样，才能成功的做成想做的事，才能在今后的道路上劈荆斩棘，而不是知难而退，那样永远不可能收获成功，收获喜悦，也永远不可能得到社会及他人对你的认可！

**七、参考文献**

殷人昆《数据结构（用面向对象方法与C++语言描述）第二版》，清华大学出版社。