**C语言项目报告**

**项目名称：\_\_\_\_\_长春地铁\_\_\_\_\_\_**

**项目成员：\_\_\_\_\_\_王德瑞\_\_\_\_\_**

**填写日期：\_\_\_\_\_\_\_\_2020/9/8\_\_\_**

1. **摘要（Abstract）**

通过邻接矩阵进行深度遍历从而能够计算在地铁线路图中的两点之间所需的最少站点，并给出具体站点及路径

使用C++实现，不同于以往用python或Java

运用深度遍历求图中的所有路径（无向图可以规避成环绕远路的情况）

用了邻接矩阵进行运算，并且根据点线关系自动生成，后来使用了更为简便的邻接表

用数组使得站点选择更简便量化

1. **问题描述（Problem Statement）**

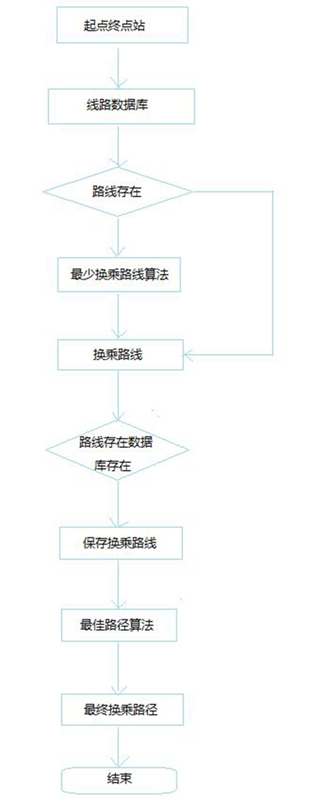
日常生活中使用长春地铁会有迷路或是对方向有疑惑，通过完整的给出各个站点从而可以使人们能够更直观的乘车以达到节省时间的目的，方便新生报到或者游客乘坐地铁时的最优路径。

根据不同的需求可以使人计算最短路径或是最短换乘。

1. **组内分工（Group Division）**

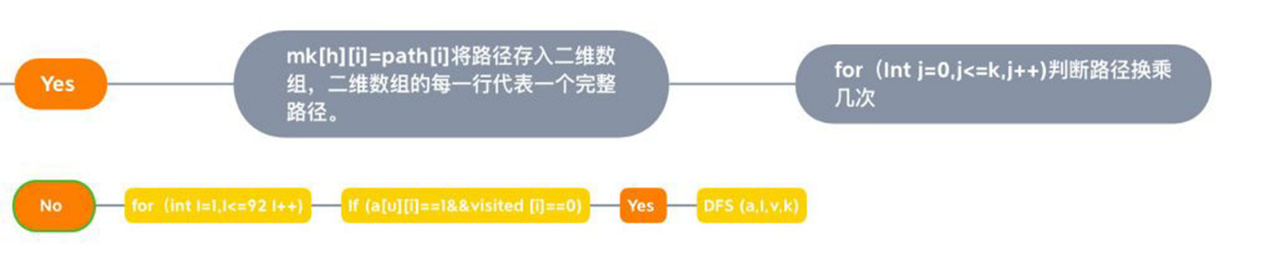
一人，具体学习了DFS算法和邻接矩阵生成，工作量大约20天，8月底进行了少量Debug和修复完善

1. **分析（Analysis）**



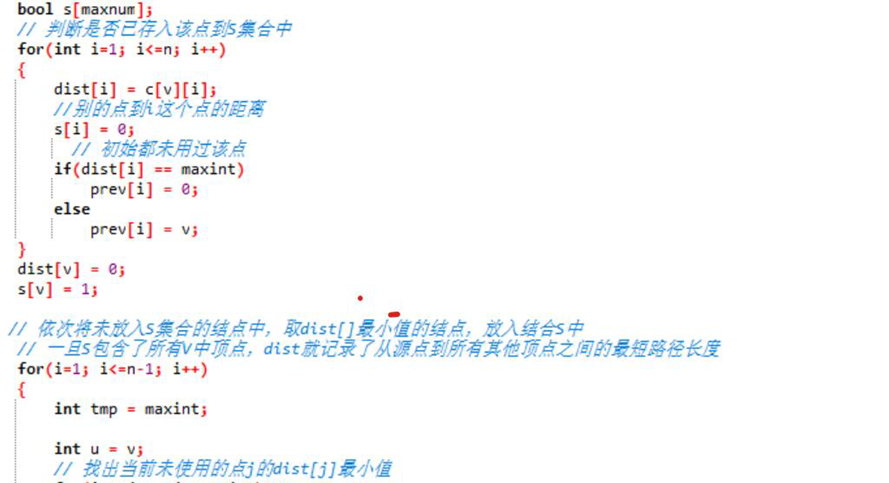
1. **设计（Design）**





1. **实施（Implementation）**

**首先定义一个bool类型的数组， 用来判断是否已经遍历先第一次遍历所有结点，得到所有结点到源点的距离， 并且将s[max num] 的初始值都设为0， 表示初始一个点都没有使用， 因为dist[max num] 的初始值都为maxint， 在第一次遍历之后， 如果dist i] 依旧为maxint的点就是与源点不相邻的，否则源点v就是此点的前一个结点， 存v到prev数组中去。prev[] =v；赋值dist[v] 的值为0， s[v] 的值为1。将源点剔除。第二次的二重遍历的目的是，依次将未放入s集合的节点中dist] 值最小的结点放入s中。当s包含了所有的顶点后， dist就记录了从源点到所有其他顶点之间的最短距离。每次循环只能标记一个顶点**

****

1. **测试（Test）**

第二个最少换乘到学校后由于环境不同容易出现不显示的问题，暂时解决情况并不理想。

最短路径经过20次以上的检验没有问题，且具有很好的移植到其他城市的优势。

