

**《数据结构》课程设计说明书**

**综合应用**

学 号： **2053182**

姓 名： **王润霖**

专 业： **信息安全**

题 号： **02**

二〇二二年八月

第一部分 题目

**1.题目描述**

上海的地铁交通网路已基本成型，建成的地铁线十多条，站点上百个，现需建立

一个换乘指南打印系统，通过输入起点站和终点站，打印出地铁换乘指南，指南内容包括起点站、换乘站、终点站。

（1）图形化显示地铁网络结构，能动态添加地铁线路和地铁站点。

（2）根据输入起点站和终点站，显示地铁换乘指南。

（3）通过图形界面显示乘车路径。

第二部分 软件功能

**2.1软件功能**

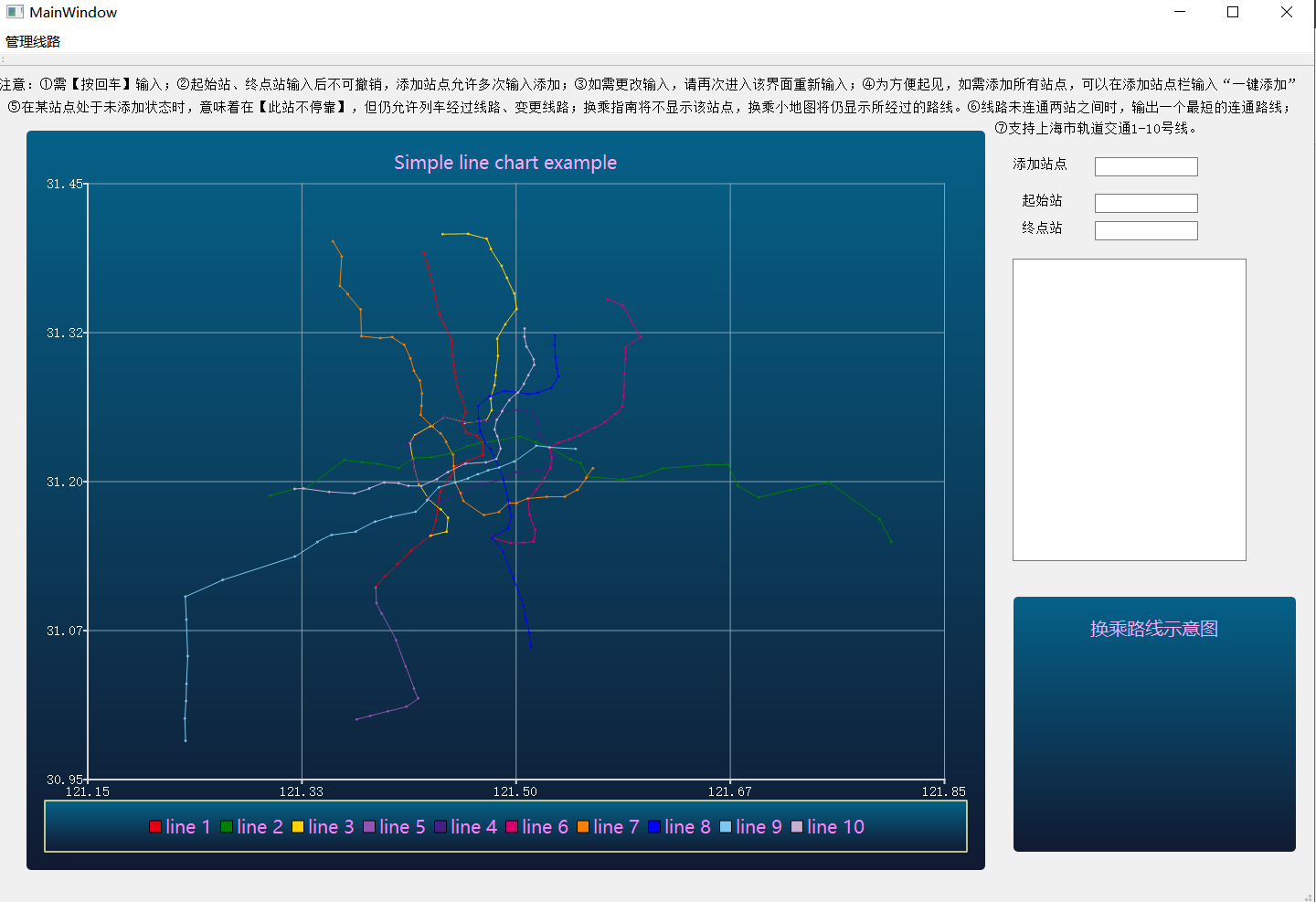
软件发布在：

<https://github.com/ChestnutSilver/DataStructureDesign>

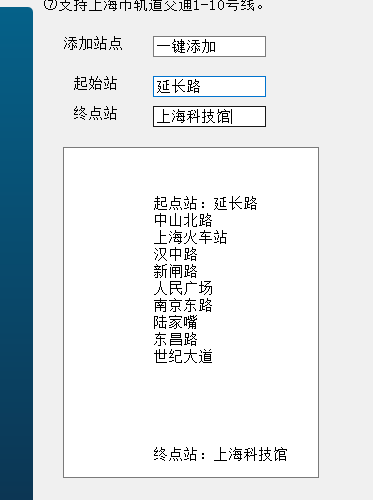
(1)设计欢迎界面；



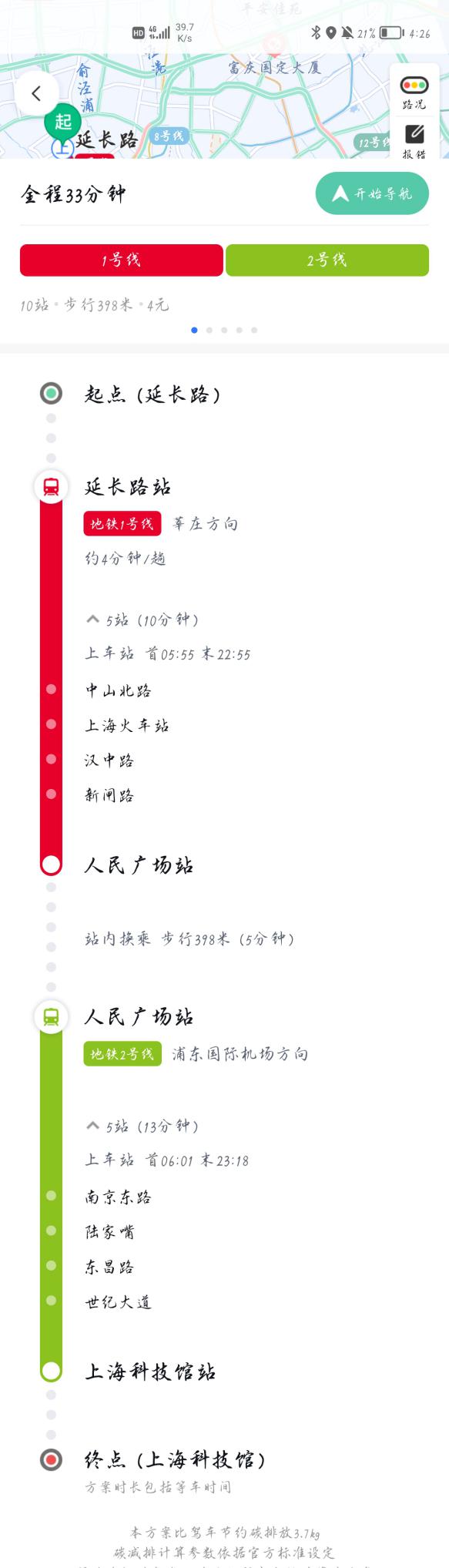
（2）图形化显示地铁网络结构，能动态添加地铁线路和地铁站点



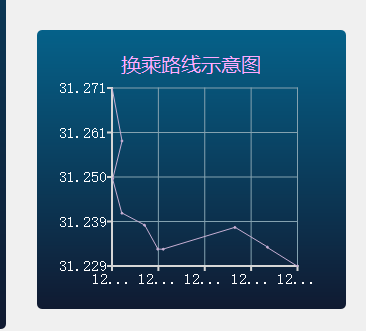
（3）根据输入起点站和终点站，显示地铁换乘指南。



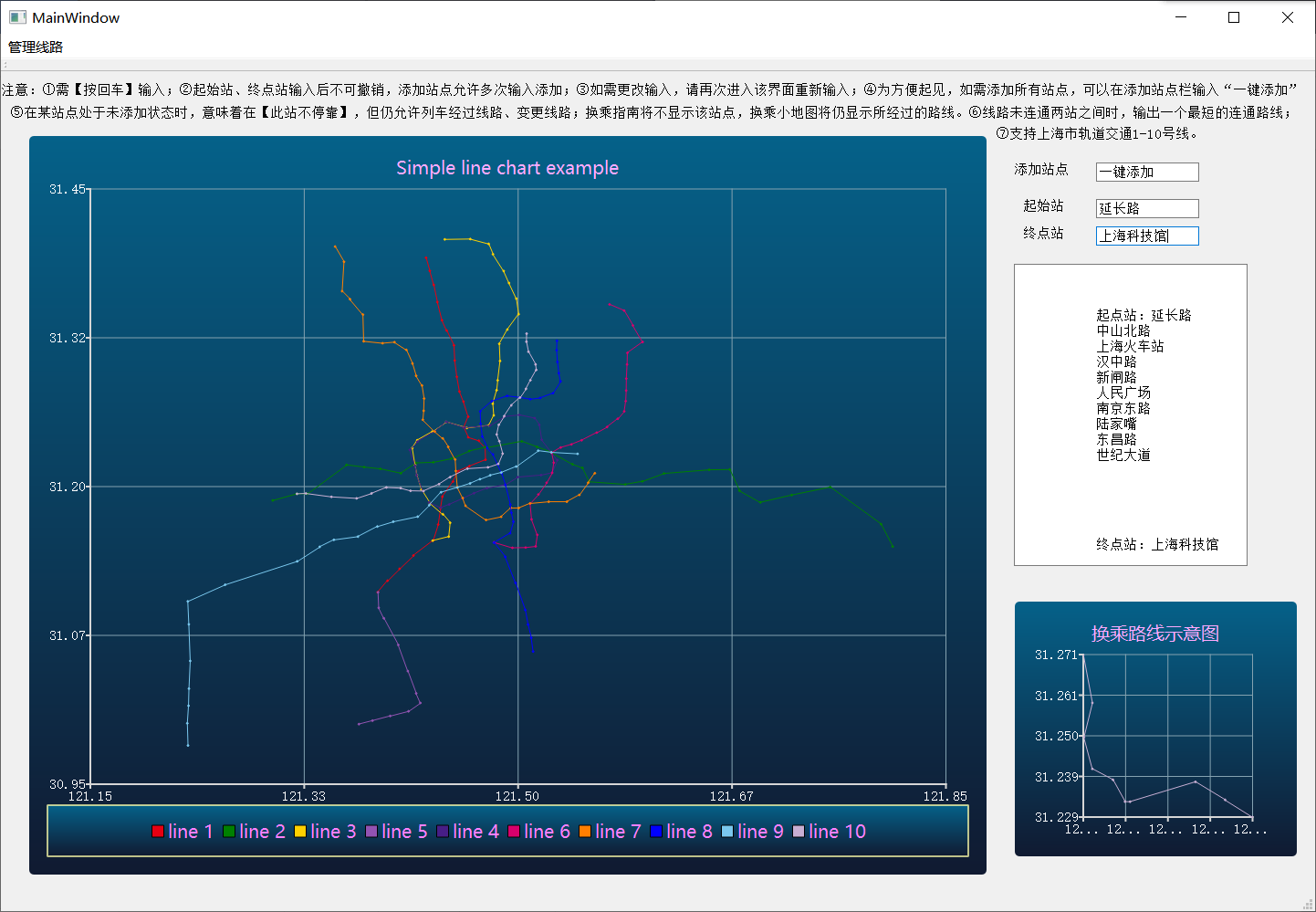
与百度地图轨道交通换乘路线一致。



（4）通过图形界面显示乘车路径。



**2.2软件样例**



第三部分 设计思想

**3.设计思想**

**涉及数据结构：**图状结构。

**相关背景：图**是较线性表和树更复杂的、描述多对多关系的数据结构。图是用于表示对象之间联结关系的抽象数据结构，通常使用**顶点**（vertex）集和**边**（edge）集进行描述：顶点表示对象，边表示对象之间的关系。根据边是否有方向和权值，图又可细分为**有向图、无向图**以及**有向网、无向网**等。当边有方向时，一条边连接的两个顶点按方向分为原点（source vertex）和终点（destination vertex）。

**使用Dijkstra算法，**（1）初始化节点，赋予每个结点路径值为一个极大值，源点的路径值为0（2）定义一个优先队列，队列中元素记录了结点的编号和节点的最短路径值，将源点压入队列。（3）当队列非空，执行以下操作：（a）u等于队头的结点，w等于队头结点的最短路径值（b）遍历u的所有边，如果能找到结点v最短路径值小于v的当前值，更新v，将v压入队列。（4）结束。

|  |
| --- |
| int MainWindow::**dijkstra**(int sec, int n)  {  memset(dist, 0x3f, sizeof dist); //初始化所有距离为最大  dist[sec] = 0; //初始化起点距离为0  for (int i = 0; i < n - 1; i++) //循环找到起点最短的且未确定最短距离的点t  {  int t = -1;  for (int j = 0; j <= n; j++)  if (!st[j] && (t == -1 || dist[t] > dist[j])) {  t = j;  }  for (int j = 0; j <= n; j++) {  if (dist[j] > dist[t] + g[t][j]) {  path[j] = t; //若从起点到点j的距离被从起点到点t，再从点t到点j更新，  //那么t就是j的前驱节点，保存到path[j]里  }  if(dist[j] < dist[t] + g[t][j])  dist[j] = dist[j];//更新最短路  else {  dist[j] = dist[t] + g[t][j];  }  }  st[t] = true; //点t的最短路径已经确定  }  if (dist[n] > 0x3f-1) return -1;  return dist[n];  } |

第四部分 逻辑结构与物理结构

**4.1逻辑结构**

**逻辑结构设计：图状结构**（无向图）

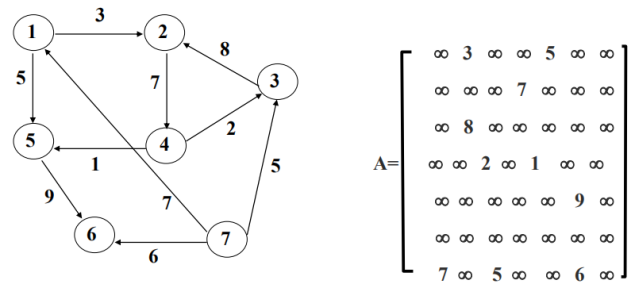
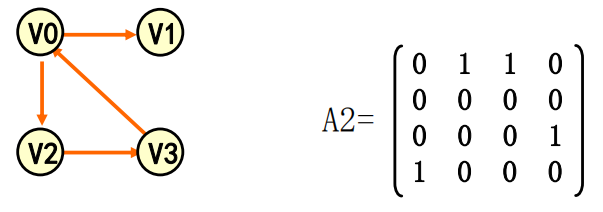
|  |
| --- |
| typedef int VertexType;  typedef struct ArcNode //表结点（边结点）  {  int adjvex; //该弧所指向的顶点的位置  struct ArcNode\* nextarc; //指向下一条弧的指针  int info;  }ArcNode; //边结点类型  typedef struct VNode //头结点  {  VertexType data; //顶点信息  ArcNode\* firstarc; //指向第一条依附该顶点的弧的指针  }VNode, AdjList[MAX\_VERTEX\_NUM];  typedef struct  {  AdjList vertices; //邻接表；存放各个顶点的数组  int vexnum, arcnum; //图的当前顶点数和边数  }ALGraph;  ALGraph G; |

**4.2物理结构**

**使用邻接矩阵：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **MGraph** | VertexType vexs[MAX\_VERTEX\_NUM]; | AdjMatrix arcs; | int vexnum | int arcnum | GraphKind kind; |
|  | 1\*MAX字节 | 4\*MAX字节 | 4字节 | 4字节 | 4字节 |

（图示以有向图、有向网为例）



第五部分 开发平台

**5.开发平台**

计算机型号 ：联想Legion Y7000 2020

处理器 Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz 2.50 GHz

机带 RAM 16.0 GB (15.9 GB 可用)

系统类型 64 位操作系统, 基于 x64 的处理器

操作系统 ：Windows 10 家庭版

开发语言 ：C++（C++11标准以上）

开发框架 ：QT

集成开发环境：Qt Version 5.12.10

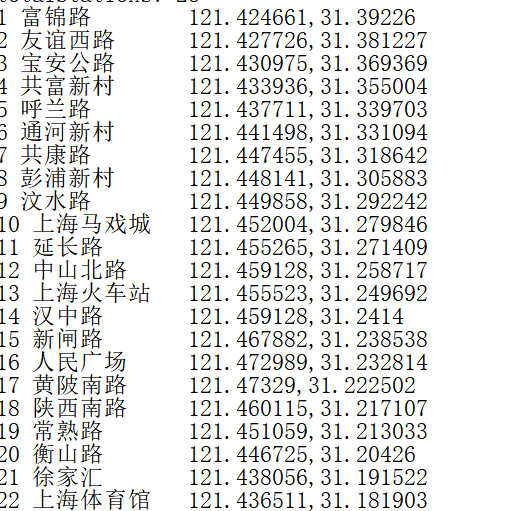
第六部分 系统的运行结果分析说明

**6.系统运行结果分析说明**

系统实现了题目要求的各项功能，并达到了较高的准确率。

通过充分的操作提示，降低了用户错误操作的可能性。

**系统使用上海地铁各站点的经纬度信息，计算结果更加精确。**

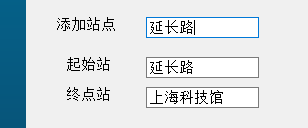


**注意：站点需要添加进地铁线路，才能在换乘指南中打印出来。**

**（1）图形化显示地铁网络结构，能动态添加地铁线路和地铁站点**

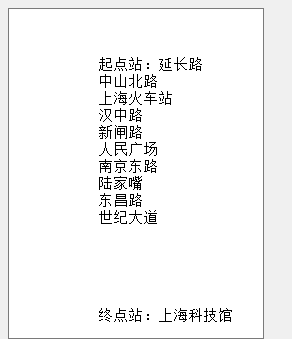
如下图所示，仅添加了部分地铁线路，线路标号在下面标注。





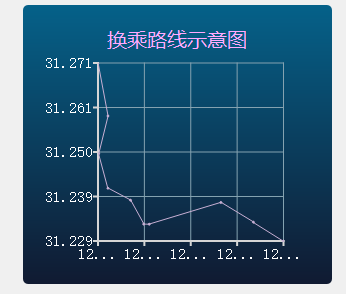
**（2）根据输入起点站和终点站，显示地铁换乘指南。**

其中，地铁换乘指南显示了地铁经过且已被添加的所有站点。



**（3）通过图形界面显示乘车路径。**

其中，横纵轴为经纬度信息。



第七部分 系统的安装与运行

**7.系统的安装与运行**

（一）打开Qt官网，找到下载目录Index of /archive/qt，选择需要下载的Qt版本，本软件版本为Qt5.12.10。



（二）选择安装组件，本软件安装组件为：



（三）已经附上所有程序执行需要的dll等文件，可以双击.exe文件运行。

第八部分 操作说明文档

程序操作简介易懂，在操作界面已有说明：

**注意:**

①需【按回车】输入；

**②起始站、终点站输入后不可撤销，添加站点允许多次输入添加；**

③如需更改输入，请再次进入该界面重新输入；

④为方便起见，如需添加所有站点，**可以在添加站点栏输入“一键添加”**

⑤在某站点处于未添加状态时，意味着在【此站不停靠】，但仍允许列车经过线路、变更线路﹔换乘指南将不显示该站点，换乘小地图将仍显示所经过的路线

，回线路未连通两站之间时，输出一个最短的连通路线;

**⑥支持上海市轨道交通1-10号线。**

第九部分 参考文献

[1]霍亚飞.Qt Creator快速入门第二版[M].北京：北京航空航天大学出版社,2014-1

[2]严蔚敏，吴伟民.数据结构（C语言版）[M].北京：清华大学出版社,2007

[3]霍亚飞，程梁. QT5 编程入门[M]. 北京：北京航空航天大学出版社,2015-1

[4]谢建平,陈治亚,邓连波,谢宜斌,杨坤.改进Dijkstra算法在大型城市轨道交通网计价系统中的应用[J].国防科技大学学报,2021,43(01):109-116.

[5]陈东银,宋艳敏.Dijkstra算法在地铁换乘中的应用[J].信息系统工程,2012(10):83-84+108.

[6]裴立秋.公交网络最优出行路径模型与算法研究[J].吉林建筑工程学院学报,2012,29(04):95-98.

[7]王天顺.大型公交网络线路查询模型与算法[J].大学数学,2010,26(04):135-142.