**数据结构与算法 第七次实验**

校园导游系统

学号：37220232203808 姓名：宋浩元

# 实验目的

1. 了解图的基础实现方法与原理，理解图的基本操作的代码编写方式
2. 学会灵活按照实际应用的存储内容要求编写图的存储结构
3. 在图的基础上进而理解编写各种最短路的基础实现方法，理解最短路决策的基本操作的代码编写方式
4. 通过实验探索不同的图的相似点与区别，发现在操作实现上的异同

# 问题描述

# 设计一个校园导游系统，为来访的客人提供各种信息查询服务。基本要求包括：

# （1）设计你所在学校的校园平面图，所含景点不少于10个。以图中顶点表示校内各景点，存放景点名称、代码、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相关信息的查询。

# （3）为来访客人提供图中任意景点的路径查询，即任意两个景点之间的一条最短的简单路径。

# （4）提供校园图中多个景点的最佳访问路线查询，即求途径多个景点的最佳（短）路径。

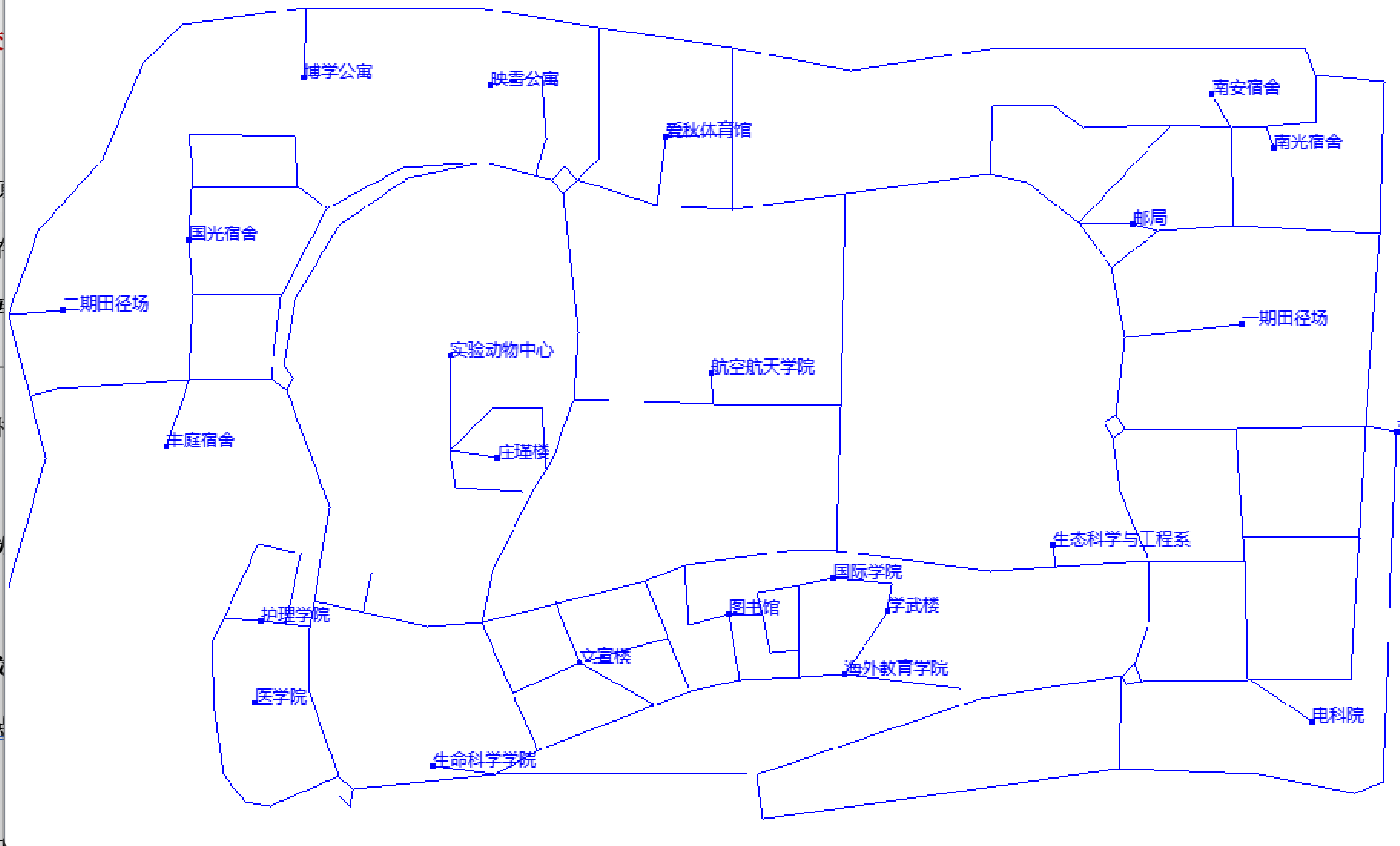
# （5）实现校园导游图的仿真界面。

# 需求分析

构造仿真界面在一张给定的连通的的每条边含有正权值的无向图中，选择起点与终点，要求寻找出包含这些点且权值和最小的通路。

如下图，对于这样的一个包含景点与道路信息的图，需要对其进行处理找出给定景点的最短路径。





# 算法设计

**1. 建筑点管理算法**

算法功能

管理建筑点的添加、删除、修改以及信息查询。

主要实现

数据操作通过 MySQL 类与数据库交互完成，涉及点的增删改查。

图形界面显示通过 drawPointAt 方法绘制建筑点，通过 drawAllPointFromSQL 从数据库加载并显示所有点。

算法流程：添加建筑点

**点击事件监听**：

鼠标右键点击地图区域，通过 mouseReleaseEvent 获取鼠标点击的位置。

检查点击位置是否已有点（通过数据库查询 get\_point\_by\_XY 判断）。

**点的添加**：

若无重复点，通过 add\_point 方法将点添加到数据库。

设置建筑点属性（如名称、描述）并更新到数据库。

**图形更新**：

在地图上动态绘制新添加的点（drawPointAt）。

重绘地图以显示最新的点和路径。

**2. 路径管理算法**

算法功能

管理景点之间的路径，包括路径绘制、删除、修改。

主要实现

路径存储在数据库中，通过 MySQL 类的 getAllLines 获取所有路径。

在地图上绘制路径（drawAllLineFromSQL 和 drawLineTo）。

鼠标事件支持动态绘制或删除路径。

算法流程：绘制路径

**点击设置起点和终点**：

鼠标左键点击两个点，分别设置起点 lastMyPoint 和终点 thisMyPoint。

**路径存储**：

通过 MySQL 插入起点与终点之间的连接关系（双向存储）。

**路径绘制**：

调用 drawLineA2B 方法，在地图上绘制连接线。

**数据库同步**：

将路径信息存储到数据库，确保数据持久化。

3. 单点间最短路径算法

算法功能

计算两个建筑点之间的最短路径并在地图上绘制。

主要实现

使用最短路径算法（如 Dijkstra 或 A\*）进行路径规划。

数据支持来自 MySQL 数据库，存储点和路径信息。

最短路径计算由 get\_min\_path 方法实现。

算法流程：最短路径计算

**输入起点和终点**：

用户通过左键点击地图，分别设置起点 startPoint 和终点 endPoint。

**路径计算**：

从数据库中读取路径信息，初始化图的邻接表。

调用 get\_min\_path 方法，使用最短路径算法计算起点到终点的路径。

**路径绘制**：

调用 showLine，在地图上逐段绘制路径。

更新地图图像并显示路径信息（如总距离）。

**多点路径规划算法**

算法功能

计算经过多个点的最佳路径（类似旅行商问题 TSP 的解法）。

主要实现

通过动态规划或近似算法计算路径。

支持用户选择多个点作为必经点（右键菜单添加）。

算法流程：多点路径规划

**用户选择待经过点**：

右键菜单选中点后，将其加入 waypoints 列表。

**路径计算**：

遍历 waypoints 列表，逐对计算相邻两点间的最短路径（调用 get\_min\_path）。

将所有路径段组合为完整路径。

**路径绘制**：

使用 find\_waypoints\_path 方法将路径绘制在地图上。

在界面上显示路径总长度和经过点顺序。

鼠标交互事件算法

算法功能

实现地图的动态交互，包括鼠标点击选择点、悬浮显示信息、右键菜单操作。

主要实现

鼠标事件主要通过 mousePressEvent、mouseMoveEvent 和 mouseReleaseEvent 实现。

鼠标悬浮时，动态显示点的信息提示框。

算法流程：鼠标悬浮事件

**捕获鼠标位置**：

在 mouseMoveEvent 中实时获取鼠标坐标 mousePos。

**点信息查询**：

调用 get\_point\_by\_XY，判断鼠标所在位置是否有建筑点。

若有，获取该点的名称和描述。

**信息框显示**：

在鼠标位置附近显示提示框，内容为建筑点信息。

若鼠标移出点范围，则隐藏提示框。

**6. 数据库操作算法**

算法功能

数据库操作支持点和路径的管理，保证系统数据与界面一致。

主要实现

数据库管理通过 MySQL 单例类实现，封装了所有增删改查操作。

点和路径信息动态同步到数据库中。

算法流程：新增点

**检查点是否存在**：

调用 get\_point\_by\_XY，判断数据库中是否已存在该点。

**插入新点**：

若不存在，调用 inser\_point 插入点信息。

**界面更新**：

调用 drawAllPointFromSQL 重新加载所有点到界面。

以下是关于 **显示建筑点相关信息** 算法的更详细分析及补充。

功能描述

**主要目标**： 当用户将鼠标悬浮在建筑点上时，动态获取该建筑点的信息（如名称、简介）并通过信息提示框展示。

**触发机制**：

鼠标移动事件 (mouseMoveEvent) 检测鼠标位置。

判断鼠标是否悬浮在建筑点上。

**关键步骤**：

根据鼠标位置查询建筑点。

若存在建筑点，动态显示建筑信息框。

若离开建筑点范围，则隐藏提示框。

算法流程

1. 鼠标悬浮事件触发

在 ScribbleArea::mouseMoveEvent 中实现：

**获取鼠标位置**：

**调用数据库查询**：

通过鼠标位置坐标，从数据库中查找是否有建筑点。

数据库查询通过 MySQL::get\_point\_by\_XY 完成，支持误差范围内的匹配。

2. 判断查询结果

**若查询结果为空**：

当前鼠标位置无建筑点。

隐藏提示框（如果提示框存在）。

释放已查询的内存（防止内存泄漏）。

**若查询结果非空**：

表示鼠标位置有建筑点。

动态显示提示框，内容包括：

**建筑名称**。

**建筑简介**。

**3. 动态显示提示框，展现建筑相关信息**

如果提示框未初始化，则创建 QLabel 作为提示框。

设置提示框样式（如背景颜色、字体等）。

设置提示框的内容和位置：

内容为查询到的建筑点信息。

提示框显示在鼠标位置的偏移处，防止遮挡鼠标光标。

提示框需考虑边界情况，防止超出地图窗口。

# 系统实现

## （1）Program Structure

本项目是一个基于 Qt 框架 实现的校园导航系统，具备用户身份验证、地图交互、路径规划和动态地图管理等功能。项目主要面向以下需求：

提供动态交互的校园地图，可查询建筑信息。

支持最短路径计算和多点路径规划。

实现用户管理，包括登录和注册。

允许管理员对地图上的点和路径进行增删改操作。

2. 项目模块划分

2.1 应用主控模块（Start、MainWindow、Welcom）

**Start 模块**：

负责初始化应用程序，管理欢迎界面（Welcom）和主窗口（MainWindow）。

提供切换界面的接口，如打开登录界面、注册界面或主导航界面。

保存当前用户信息（如用户名和是否为管理员）。

**主要功能**：

open\_mainwindow()：启动主窗口。

open\_login() 和 open\_sigin()：分别打开登录和注册界面。

set\_user(QString name)：设置当前登录的用户名。

isAdmin()：判断当前用户是否为管理员。

**Welcom 模块**：

欢迎界面，显示项目简介及系统入口按钮（登录和注册）。

**MainWindow 模块**：

应用程序的核心界面，整合地图交互区（ScribbleArea）和管理工具。

提供用户访问地图、路径规划和动态修改地图内容的功能。

2.2 地图交互模块（ScribbleArea、PaintArea）

**ScribbleArea 模块**：

地图的核心绘图区域，支持地图渲染、用户交互和动态绘制点、线。

支持鼠标操作：

**左键**：选择起点或终点。

**右键**：打开上下文菜单，允许添加、删除或修改地图上的点和路径。

**鼠标悬浮**：显示建筑点的详细信息。

**主要功能**：

drawPointAt()：绘制地图上的点。

drawLineA2B()：绘制两点之间的路径。

find\_waypoints\_path()：计算并绘制多点之间的最优路径（旅行商问题）。

mouseMoveEvent()：动态检测鼠标位置，显示建筑信息提示框。

**PaintArea 模块**：

基础绘图功能，负责绘制形状和路径。部分功能与 ScribbleArea 重叠，可优化整合。

2.3 图与路径规划模块（LibPoint、Point）

**LibPoint 模块**：

自定义图结构，基于邻接矩阵实现点与路径的管理。

提供最短路径计算功能，主要使用 **Dijkstra 算法**。

**主要功能**：

get\_min\_path(QPoint A, QPoint B)：计算两点之间的最短路径。

append\_edge()：在图中添加路径（边）。

init\_GMat()：初始化图的邻接矩阵。

**Point 模块**：

封装地图上的点，包含点的坐标、名称、描述等信息。

提供坐标与字符串的转换方法。

2.4 数据管理模块（MySQL）

**MySQL 模块**：

基于 SQLite 的数据库管理模块，负责用户信息、地图点和路径数据的存储和管理。

表结构：

POINT 表：存储地图上的点，包括坐标、名称和类型（普通点/建筑点）。

NEAR 表：存储两点之间的连接信息（路径）。

**主要功能**：

get\_point\_by\_XY()：根据坐标获取点的信息。

inser\_point() 和 remove\_point\_by\_XY()：添加和删除点。

inser\_near()：添加路径。

getAllPoints() 和 getAllLines()：加载所有点和路径数据。

2.5 用户管理模块（Login、Sigin）

**Login 模块**：

提供用户登录功能，根据数据库验证用户名和密码。

登录成功后判断用户是否为管理员，并打开相应的功能界面。

**Sigin 模块**：

提供用户注册功能，支持基本输入验证，如用户名长度、密码确认等。

2.6 UI 文件

多个 .ui 文件用于定义界面布局，包括登录界面、注册界面、主窗口、地图等。所有 .ui 文件由 Qt Designer 设计，加载到对应的类中。

3. 程序运行流程

**程序启动**：

由 Start 初始化 Welcom 界面，用户选择“登录”或“注册”。

**用户登录或注册**：

登录：调用 Login 模块，验证用户信息。

注册：调用 Sigin 模块，创建新用户。

**加载主界面**：

登录成功后，切换到 MainWindow，加载地图数据。

用户可通过 ScribbleArea 交互：

查询建筑信息。

添加、修改或删除点和路径。

计算最短路径或多点最优路径。

**地图操作**：

通过鼠标事件在地图上动态操作，绘制或更新点和路径。

数据实时同步到数据库。

**程序退出**：

释放所有动态分配的资源。

## （2）Function Implementation

**1**. 主控模块功能实现

1.1 Start 模块

**文件：start.cpp、start.h**

功能描述**：**

**作为应用程序的入口点，负责初始化和管理全局对象（如主窗口、欢迎界面）。**

**提供方法切换界面（欢迎界面、登录界面、注册界面、主界面）。**

**记录当前登录用户的信息。**

主要方法**：**

**open\_mainwindow()：打开主窗口并加载地图。**

**open\_login() 和 open\_sigin()：分别打开登录界面和注册界面。**

**set\_user(QString name)：设置当前登录用户。**

**isAdmin()：判断当前登录用户是否为管理员（通过检查用户名）。**

1.2 Welcom 模块

**文件：welcom.cpp、welcom.h**

功能描述**：**

**提供欢迎界面，用户可以通过按钮进入登录或注册界面。**

**提供对窗口大小的动态调整（resizeEvent）。**

主要方法**：**

**on\_sigin\_buttom\_clicked()：进入注册界面。**

**on\_login\_button\_clicked()：进入登录界面。**

1.3 MainWindow 模块

**文件：mainwindow.cpp、mainwindow.h**

功能描述**：**

**主程序界面，集成地图、路径管理和其他功能。**

**包含 ScribbleArea，作为地图的交互区域。**

**提供清空所有地图数据的功能。**

主要方法**：**

**clearAllData()：清空地图上的所有点和路径数据。**

2. 地图交互功能实现

2.1 ScribbleArea 模块

**文件：scribblearea.cpp、scribblearea.h**

功能描述**：**

**地图绘制和交互的核心区域，支持用户操作（如鼠标点击、悬浮、右键菜单）以及地图渲染（点、线）。**

**支持动态绘制点和路径、计算最短路径以及多点路径规划。**

主要方法**：**

**drawPointAt(const QPoint &pos)：在指定位置绘制点。**

**drawLineA2B(const QPoint &startPoint, const QPoint &endPoint)：绘制两点之间的路径。**

**find\_waypoints\_path()：计算并绘制经过多个点的最优路径（旅行商问题）。**

**mousePressEvent(QMouseEvent \*event)：**

**鼠标左键：设置起点或终点。**

**鼠标右键：弹出上下文菜单，进行动态地图管理。**

**mouseMoveEvent(QMouseEvent \*event)：检测鼠标悬浮的建筑点，动态显示建筑信息提示框。**

**showLine(QPoint A, QPoint B)：根据起点和终点，计算并绘制最短路径。**

2.2 PaintArea 模块

**文件：paintarea.cpp、paintarea.h**

功能描述**：**

**负责绘制形状和路径的基础组件。**

**与 ScribbleArea 功能有所重叠，但更通用，适合绘制非地图相关内容。**

3. 图与路径规划功能实现

3.1 LibPoint 模块

**文件：libpoint.cpp、libpoint.h**

功能描述**：**

**自定义图数据结构，支持路径管理（添加、删除、查询）和最短路径计算。**

**使用邻接矩阵表示图结构，并实现了基于 Dijkstra 算法的路径计算。**

主要方法**：**

**get\_min\_path(QPoint A, QPoint B)：**

**使用 Dijkstra 算法计算两点之间的最短路径。**

**返回路径经过的所有点。**

**append\_edge(int x, int y, int nx, int ny)：添加一条路径（边）。**

**init\_GMat(QVector<QPoint> P)：根据输入点集合初始化邻接矩阵。**

**remove\_edge()：删除指定路径。**

3.2 Point 模块

**文件：point.cpp、point.h**

功能描述**：**

**封装地图上的点，包含以下信息：**

**坐标（x，y）。**

**点类型（普通点/建筑点）。**

**名称和描述（如建筑名称和简介）。**

**提供点的序列化和基本转换方法。**

主要方法**：**

**toQPoint()：将点对象转换为 QPoint。**

**to\_string()：将点信息序列化为字符串，便于显示和存储。**

4. 数据管理功能实现

4.1 MySQL 模块

**文件：mysql.cpp、mysql.h**

功能描述**：**

**负责与 SQLite 数据库交互，管理用户信息、地图点和路径的存储和操作。**

**支持地图数据的动态更新和持久化存储。**

主要方法**：**

**get\_point\_by\_XY(Point \*\*point, int x, int y)：**

**根据坐标查询点信息，并返回最近的点。**

**inser\_point(Point \*point)：将新的点插入数据库。**

**remove\_point\_by\_XY(int x, int y)：删除指定坐标的点。**

**inser\_near(int x1, int y1, int x2, int y2)：插入两个点之间的路径。**

**getAllPoints()：从数据库加载所有点。**

**getAllLines()：从数据库加载所有路径。**

5. 用户管理功能实现

5.1 Login 模块

**文件：login.cpp、login.h**

功能描述**：**

**提供用户登录功能，通过数据库验证用户名和密码。**

**根据用户名判断用户权限（普通用户或管理员）。**

主要方法**：**

**on\_login\_confirm\_clicked()：验证用户信息并登录。**

**on\_login\_cancel\_clicked()：关闭登录界面。**

5.2 Sigin 模块

**文件：sigin.cpp、sigin.h**

功能描述**：**

**提供用户注册功能，支持基本输入验证（如用户名长度、密码确认）。**

**注册成功后直接登录。**

主要方法**：**

**on\_sigin\_confirm\_clicked()：完成用户注册并写入数据库。**

**on\_sigin\_cancel\_clicked()：关闭注册界面。**

6. 功能实现流程

6.1 用户操作流程

**用户启动程序，进入 Welcom 界面。**

**用户选择“登录”或“注册”：**

登录**：验证用户名和密码后进入 MainWindow。**

注册**：完成注册后进入 MainWindow。**

**用户在 MainWindow 进行地图交互：**

**查询建筑信息。**

**添加、修改或删除地图上的点和路径。**

**计算最短路径或规划多点路径。**

6.2 地图管理流程

**用户在地图上点击：**

**左键设置起点或终点。**

**右键打开上下文菜单，选择操作（如新增点、删除点、连接路径）。**

**ScribbleArea 将用户操作同步到数据库（通过 MySQL 模块）。**

**更新后的点和路径通过 LibPoint 和 ScribbleArea 渲染到地图上。**

6.3 路径规划流程

**用户设置起点和终点，触发 showLine() 方法。**

**调用 LibPoint::get\_min\_path() 计算最短路径。**

**路径结果绘制在 ScribbleArea 上。**

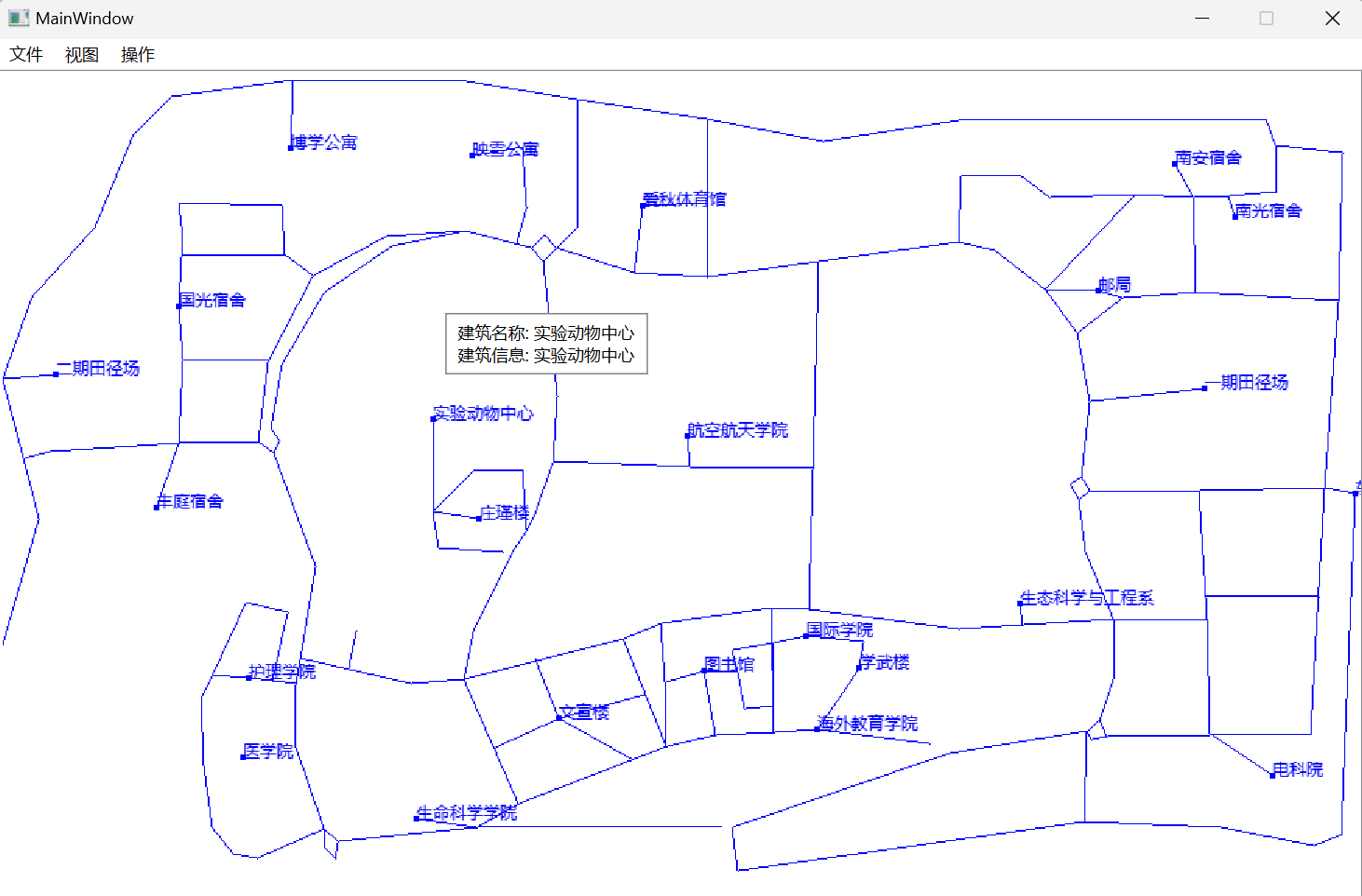
# 测试分析

**为了方便老师验收，我将项目打包成了.exe文件放在了release文件夹下，但是目前程序还有一定的bug会导致程序崩溃（在登录后），避免的方法我测试了有几种方法：**

1. **刚登录进去不要移动鼠标，等待5秒以上。**
2. **接着可以尝试向上/下开始慢慢的画圈，如果画圈时没有出现崩溃即可正常使用。**

**经过附在文件夹中的代码的测试得到的结果如下，经检验是正确的。**

**功能一：当鼠标移动到建筑点时会出现信息框，有名字和简介（简介为了方便我暂时设置成一样的了）。**



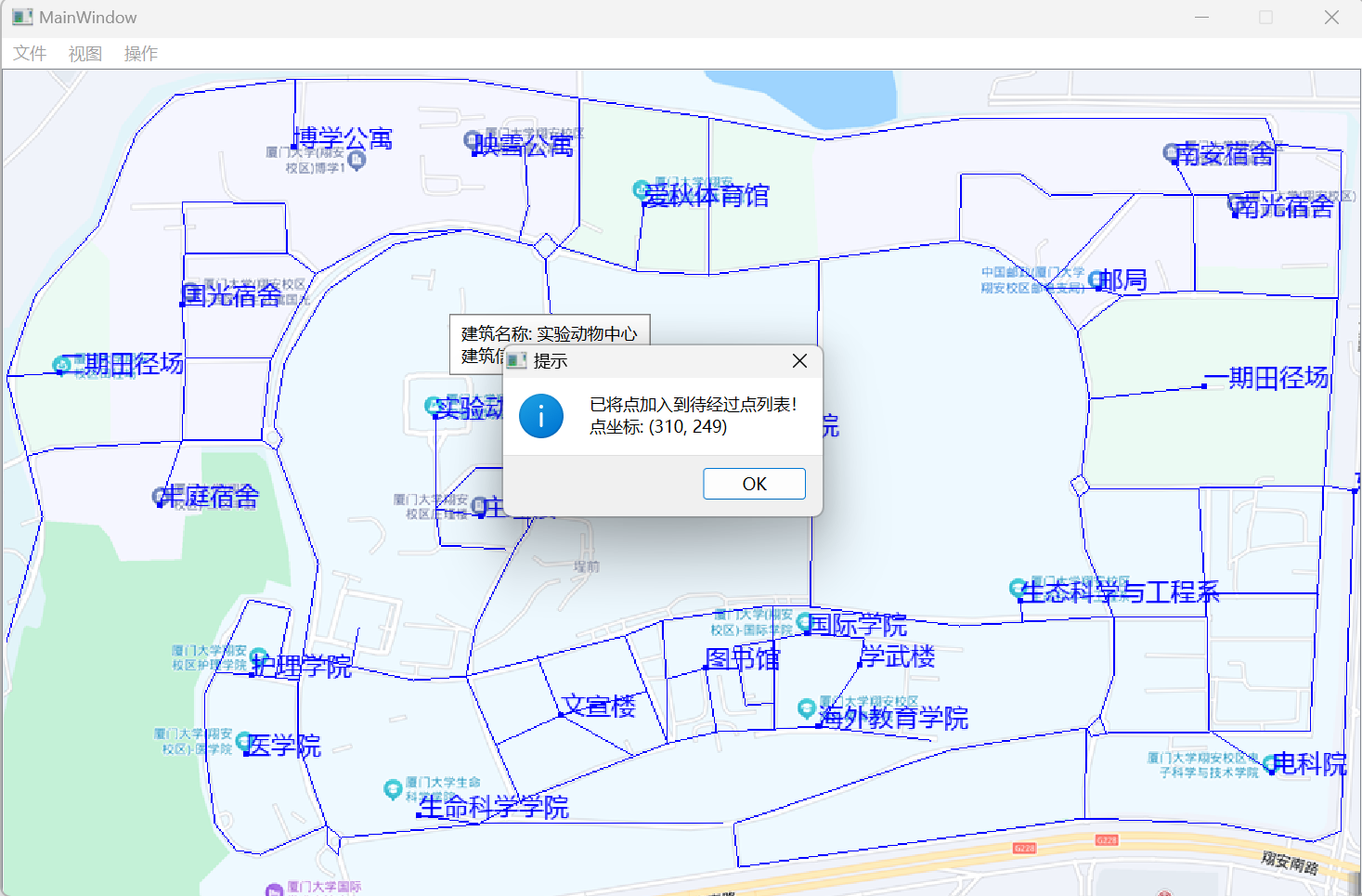
**功能2：设置起点**



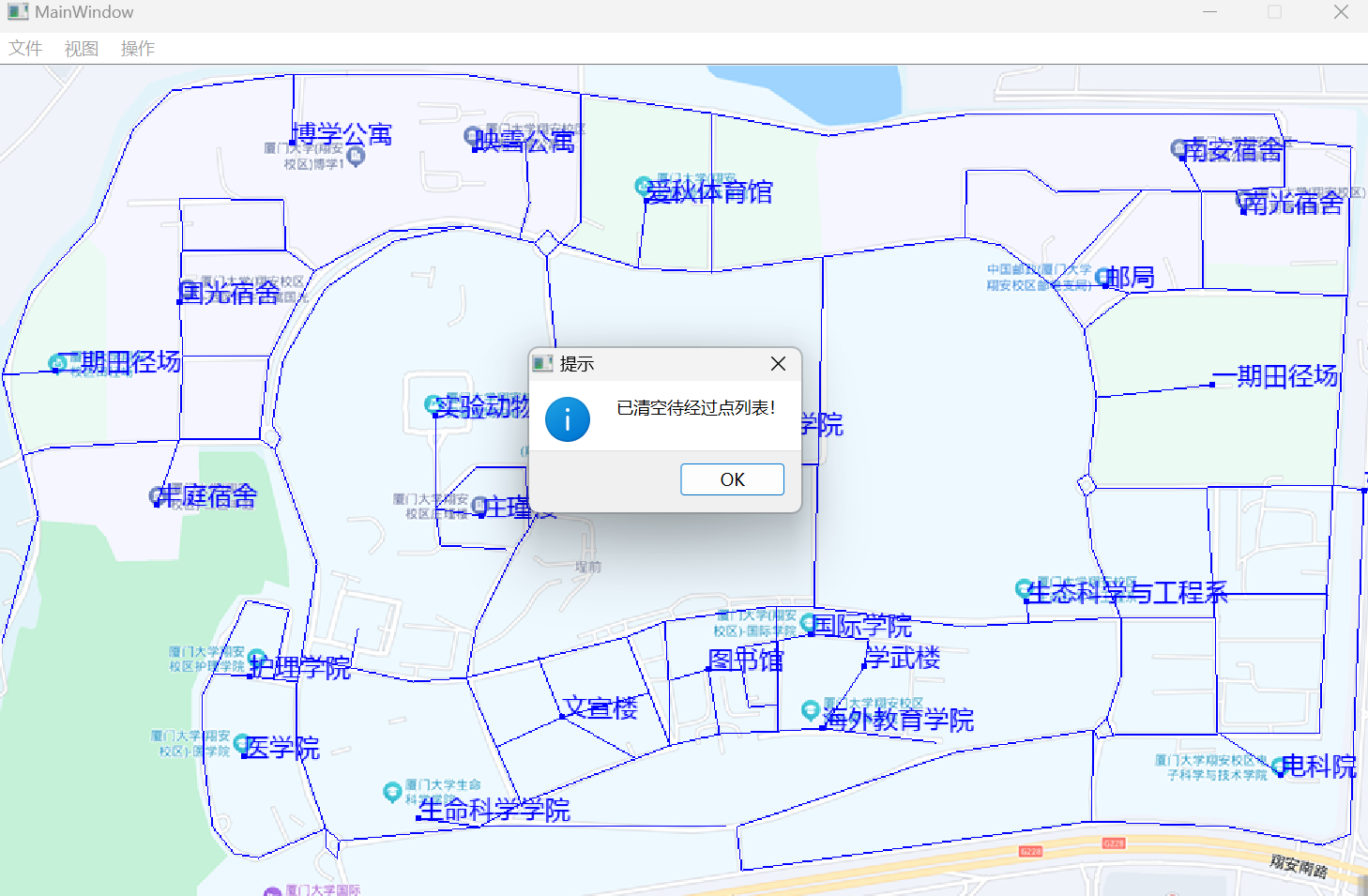
**功能三：设置终点**



**功能四：将景点加入待经过路径。**

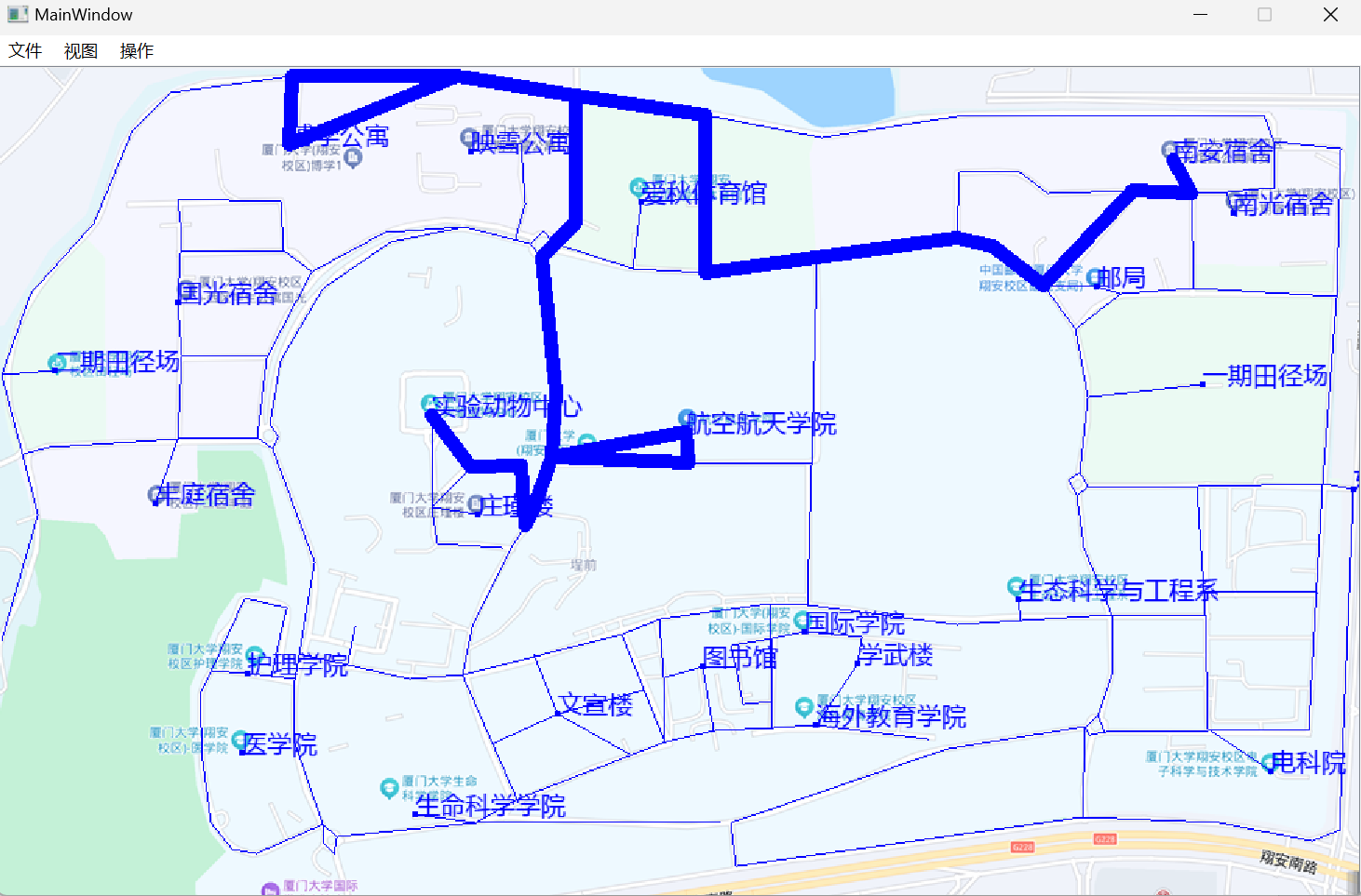


**功能五：清空待经过路径**



**功能六：求待经过点的最短路径**

**依次经过的时实验动物中心，航空航天学院，博学公寓，南光公寓。**



**功能七（admin有更多的权限）主要是对地图的构建和管理**



# 六、实验小结（即总结本次实验所得到的经验与启发等）：

在本次实验中，我尝试具体运用了图的操作，还完成了qt项目的构建和简单数据库操作的具体应用，在实体机的实验中我能够更深刻地理解对这一部分数据结构的执行方式与特点，并且在编写代码的过程中，我通过不断的调试去寻找语句之间的问题和不足，在潜移默化中提高了我的代码编写能力，这是一次完成效果良好的实验！