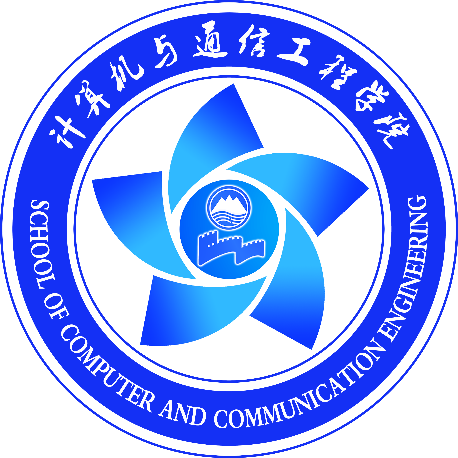
****

**东北大学秦皇岛分校**

**计算机与通信工程学院**

**数据结构课程设计**

## **设计题目** 校园导游咨询系统

目录

[**设计题目** 校园导游咨询 0](#_Toc90674082)

[课程设计任务书 2](#_Toc90674083)

[设计题目： 2](#_Toc90674084)

[一、设计实验条件 2](#_Toc90674085)

[二、设计任务及要求 2](#_Toc90674086)

[三、设计报告的内容 2](#_Toc90674087)

[课程设计报告 4](#_Toc90674088)

[前言 4](#_Toc90674089)

[设计主体 4](#_Toc90674090)

[3.1需求分析 4](#_Toc90674091)

[3.2系统设计 5](#_Toc90674092)

[3.3系统实现 5](#_Toc90674093)

[3.4用户手册 8](#_Toc90674094)

[3.5测试 12](#_Toc90674095)

[结束语 12](#_Toc90674096)

# 课程设计任务书

### 设计题目：

### 一、设计实验条件

计算机与通信工程学院实验室

### 二、设计任务及要求

任务：设计一个校园导游程序，为来访的客人提供各种信息查询服务。

要求：

（1）设计校园平面图，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、简介等信息；以边表示路径，存放路径长度的相关信息。

（2）为来访客人提供图中任意景点的相关信息查询。

（3）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一条最短路径。

（4）提供图中任意景点问路查询，即求任意两个景点之间的所有路径。

（5）校园导游图的景点扩充功能。

（6）道路信息，如道路类别（车道、人行道等）、沿途景色等级。

### 三、设计报告的内容

1. **设计题目与设计任务（设计任务书）**
2. **前言（绪论）(设计的目的、意义等)**
3. **设计主体（各部分设计内容、分析、结论等）**

**3.1需求分析**

以无歧义的陈述说明程序设计的任务，强调的是程序要做什么？给出功能模块图和流程图。同时明确规定：输入的形式和输出值的范围；输出的形式；程序所能够达到的功能；测试数据：包括正确的输入及其输出结果和含有错误的输入及其输出结果。

**3.2系统设计**

说明本程序中所有用到的数据及其数据结构的定义，包含基本操作及其伪码算法。画出函数之间的调用关系图；写出主程序及其主要模块的伪码流程。

**3.3系统实现**

给出算法的实现；程序调试过程中遇到的问题是如何解决的；对设计与实现的回顾和分析；算法的时空分析和改进思想。

**3.4用户手册**

说明任何使用你编写的程序，详细列出每一步的操作步骤。

**3.5测试**

给出测试过程及结果。

1. **结束语（设计的成果，展望等）**
2. **参考资料**
3. **附录**

带注释的源程序。

**四、设计时间与安排**

**1、设计时间：1周**

**2、设计时间安排：**

熟悉实验设备、收集资料：3 天

设计图纸、实验、计算、程序编写调试：14 天

编写课程设计报告：3 天

答辩：1 天

# 课程设计报告

### 前言

设计目的：运用数据结构和C++面向对象知识解决实际问题，在实际项目中锻炼自己编程、设计、分析、解决问题的能力。通过实现校园导游咨询图形化系统的开发，让所学知识得到运用的同时赋予这个软件一定的实用性，从而达到软件设计的目的。

意义：数据结构是计算机专业的核心课程，是计算机科学的算法理论基础和软件设计的技术基础。数据结构是实践性很强的课程。课程设计提升自己实践能力的一个强有力手段，在这个过程中，以下能力得到锻炼：

1.了解并掌握数据结构与算法的设计方法，具备初步的独立分析和设计能力；

2.初步掌握软件开发过程的问题分析、系统设计、程序编码、测试等基本方法和技能；

3.提高综合运用所学的理论知识和方法独立分析和解决问题的能力；

4.训练用系统的观点和软件开发一般规范进行软件开发，培养软件工作者所应具备的科学的工作方法和作风。

### 设计主体

### 3.1需求分析

1. 设计景点信息代号，以边表示路径，存放路径长度（实际实现时是根据点的坐标自动计算路径长度），显示方位。
2. 搜索框能搜索景点、建筑信息，输入部分文字即可自动匹配，点击景点可在地图高亮显示。
3. 路线查询，提供出发点到目的地的所有简单通路，并且显示沿途风景等级的均分，路径长度信息。
4. 路径查询结果自动排序，可以按照综合排序、路径最短优先、景色最佳优先进行排序。
5. 添加建筑、景点、关节点。
6. 显示人行道（不包含车道附近的人行道），车行道
7. 设置景色权重，0~10分，含道路权重和景点建筑权重，最终得分为平均值，将在搜索路径时显示

### 3.2系统设计

本软件在设计阶段计划使用QT开发，完成上述功能。首先明确功能，并设计了软件主窗口外观。



图表 1软件主窗口外观设计

然后，QT设计师界面中插入的背景图中标记校园关节点、建筑、景点坐标，输入节点相关信息（景色评分、节点名、类型）即可完成初始化。

接下来是路径的初始化、路径的初始化需要定义各个点之间的关系（包括道路类型、道路的景色评分）、于是将这些关系写入文件、确认无误后进行静态编译（之后运行不再需要这个文件）。

以上完成了对图的初始化操作，接下来是功能实现部分的设计。

由于每个点的坐标在初始化时可以通过程序自动获得、于是在设计路径长度时采用根据坐标计算长度的方式，这个长度值在边的初始化过程就计算出来，并保存在边对象中。之后在实现路径搜索功能时可以直接利用。

### 3.3系统实现

本软件是基于Qt框架开发的，运用了Qt提供的类库，下面仅介绍自己设计的类。

MainWindow类：

用于图形界面和内部逻辑的交互（即前后端交互，通过定义信号和槽的连接实现），在构造函数中对Graph类进行初始化。

公有函数：



成员变量：



1. Graph类：

定义了和图有关的函数和变量，变量主要包括Vertex类对象和Edge类对象。

成员变量：



公有函数：



私有函数：





Vertex类：

定义了节点有关变量和函数。主要包括坐标、节点类型、景色权重等信息。

成员变量：



公有函数：



私有函数：



1. Edge类：

定义了和边信息的变量和函数，主要包括路径类型、长度、路名等。

成员变量：

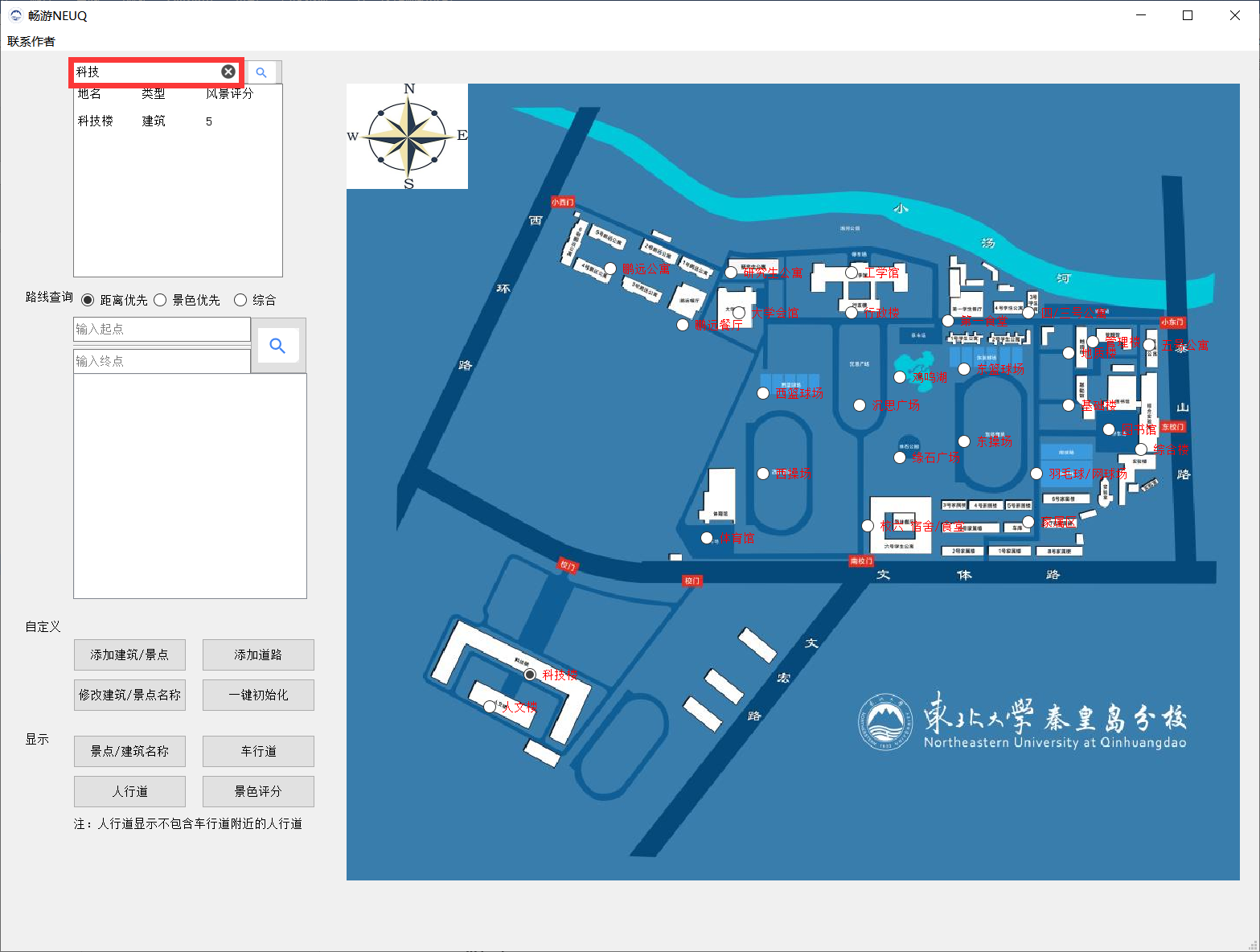


公有函数：



### 3.4用户手册

##### 1、景点搜索：

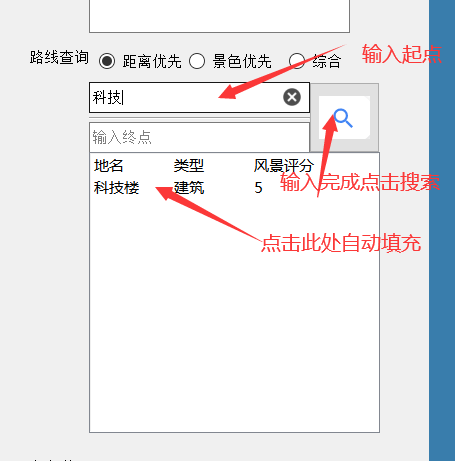


图表 2 在搜索框输入

键入部分字后，程序自动匹配符合的关键字，点击下方选项即可完成搜索；若关键字有误，下方文本框不会有提示信息，此时若点击搜索按钮不会有作用。

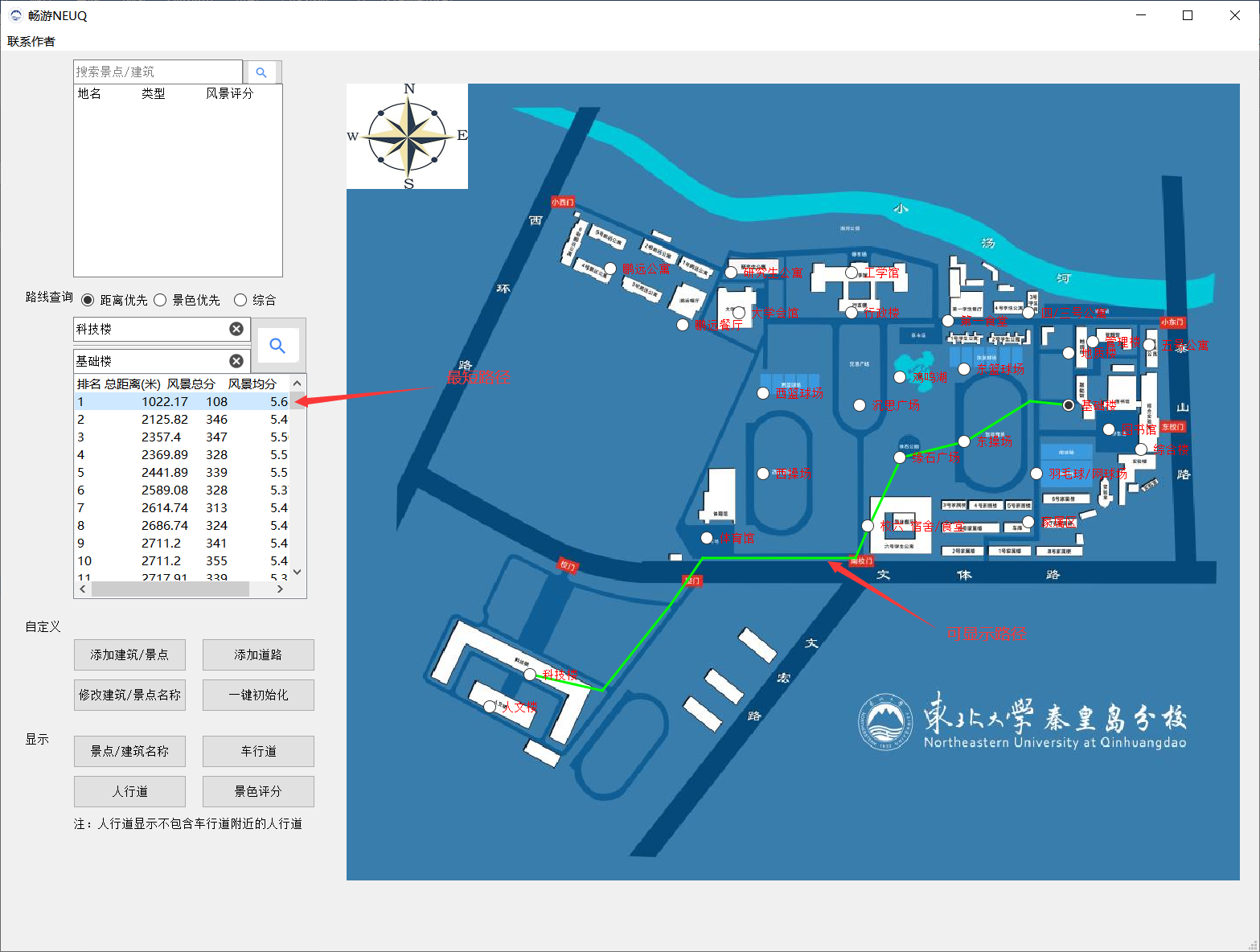
##### 2、路径搜索功能

使用路径搜索时有三个选项分别是：距离优先、景色优先、综合。默认选择距离优先。首先点击“输入起点”框，完成输入，输入时同样会自动匹配，若输入有误（即匹配不到）会要求重新输入。当弹出自动匹配的地点后，点击相应按钮可完成自动填充；若输入不完整就点击搜索会弹出提示，要求检查输入。



图表 3 路径搜索步骤

点击搜索后，根据查询条件不同，显示的路径不同。以距离优先为例，排名第一的路径一定是距离最短的，其他路径是根据算法计算出来的。（值得注意的是，这里显示的路径并不是两点之间的所有路径，原因是在实现这个功能时发现设计时选取的点有大概五十多个，而两点之间的路径条数若不加限制，其数量级是阶乘级的，常数时间内无法完成运算，后面测试中会详细提到。）



图表 4 路径搜索结果和效果

##### 3、添加景点、建筑或关节点

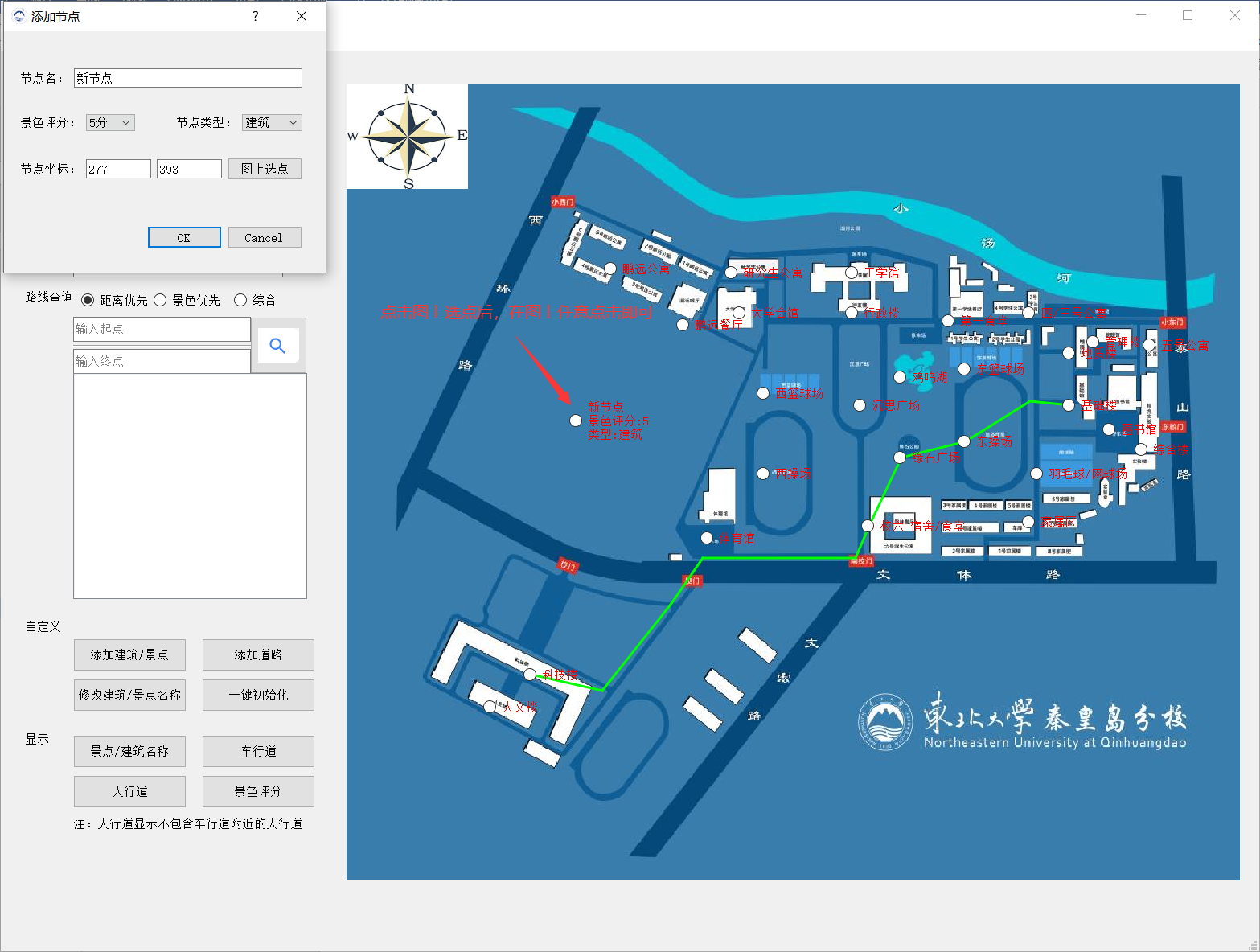
点击添加景点或建筑按钮



图表 5 添加景点、建筑或关节点按钮



图表 6 添加节点步骤



图表 7 添加节点步骤2

##### 添加道路、修改建筑\景点名称

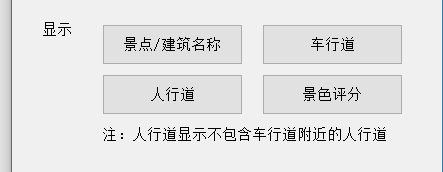
这个两个功能和前面一个功能类似，由于时间有限，故没有实现。

##### 一键初始化

该功能正常情况下无需点击，若出现添加节点错误之类的误操作可以点击进行一键还原。

##### 信息显示

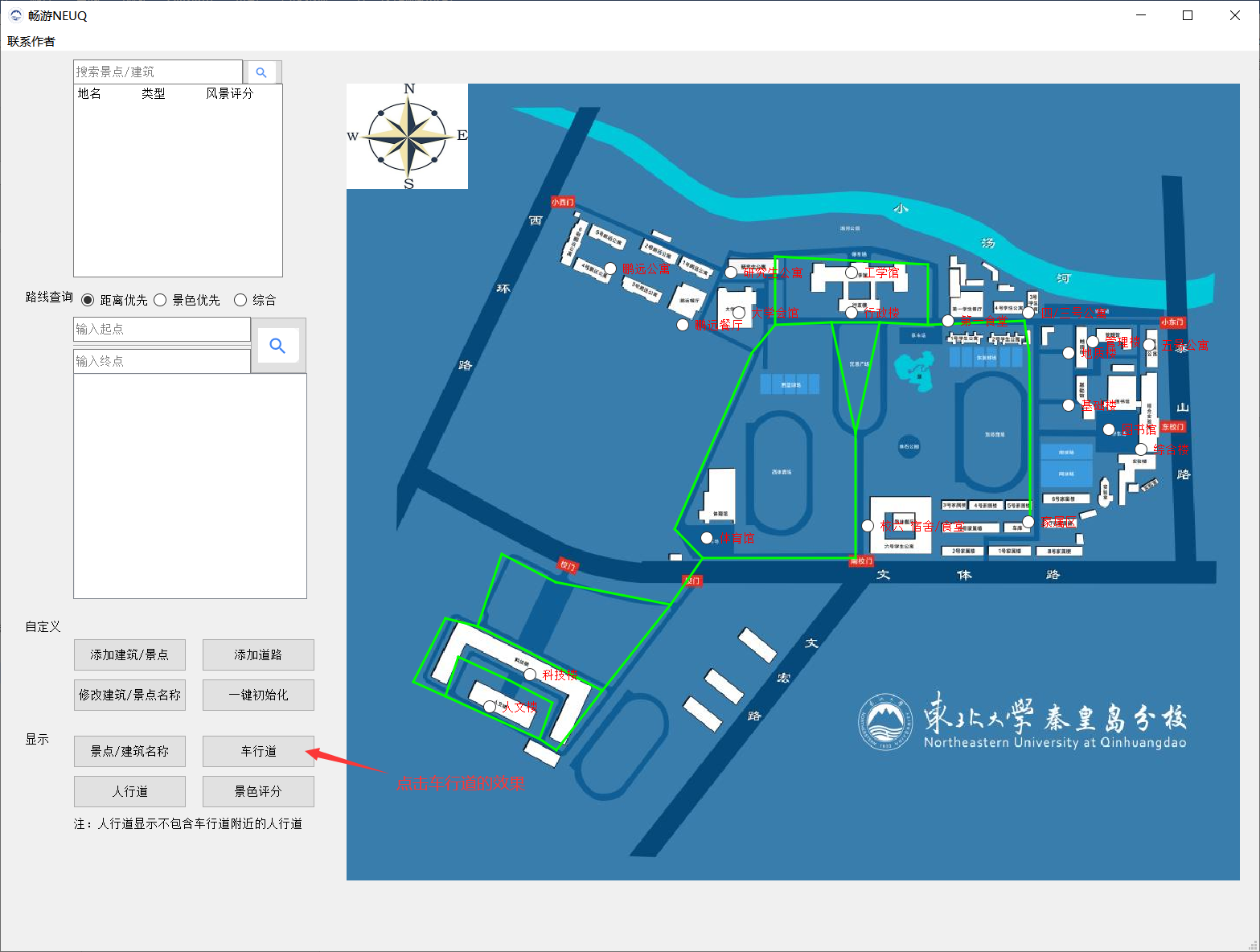
信息显示包括以下内容：



图表 8 信息显示包括的功能

其中“景点/建筑名称“按钮点击第一下只显示景点、点击第二下只显示建筑，第三下二者都显示、第四下又只显示景点、以此类推。

其余三个功能，都是点一下显示、再次点击关闭。



图表 9 点击车行道的效果

### 3.5测试

经过大量测试，本程序有着良好的健壮性，但是存在以下不足：

两点之间所有路径搜索无法完成。

由于工程量过大，设计模式不够完善，导致功能添加实现非常繁琐。

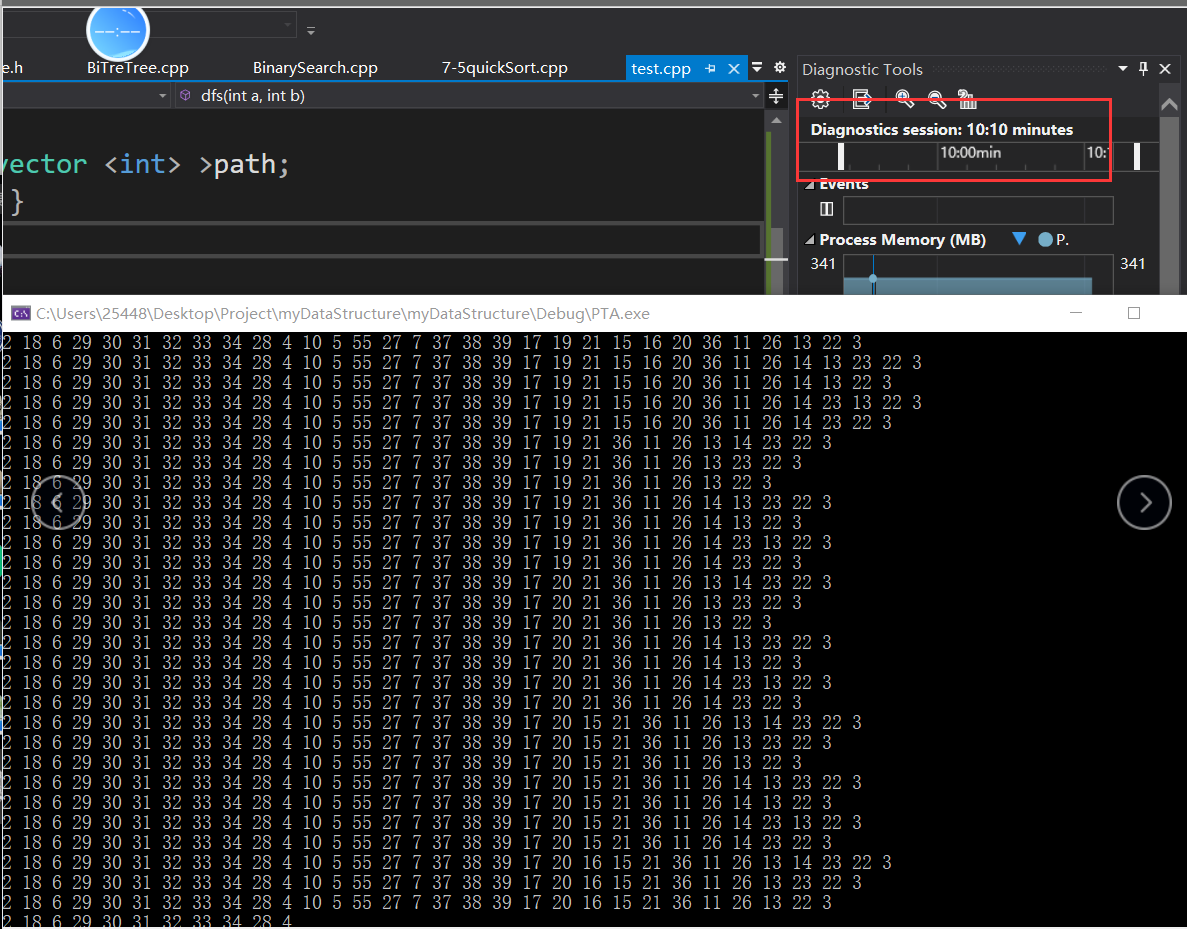
点位的选择上不够完善，只选取了部分关节点（因为关节点和路径的设置都需要人为完成工作量大）。导致路径显示上有所偏差

### 结束语

通过这个项目的实践本人感悟颇深，深刻明白了设计模式和数据结构的重要性，若一个程序设计不够完善，这个工程的可扩展性就差、不易于维护且代码会产生冗余。而如果没有数据结构作为基础，这个程序运行的效率就低，或者是时间复杂度或空间复杂度太高。

例如，在设计计算两点之间所有路径的算法时，我由于没有考虑时间复杂度的因素，以为是算法写错了，为此查阅资料，确认算法无误，单独抽出算法逻辑也是可以运行的，但是运行大量数据的时候便无法求出结果。究其原因是，图是全连通，任意简单路径，每增加一个结点，原有路径都可以在各个结点上绕一下，时间复杂度显然是n!。如果不是全连通，也不会影响数量级，而我选取的节点数有接近六十个，显然在常数时间是不可能完成的（我尝试跑了十分钟这个程序）。

总而言之，数据结构和算法是一个程序的灵魂所在，好的程序一定离不开好的数据结构和算法上的设计。在今后的学习生活中，一定还要不断提高数据结构和算法的运用能力。



图表 10 尝试遍历两点间所有路径

**参考资料**

1. 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
2. 谭浩强．C++程序设计（第3版）[M] ．背景：清华大学出版社，2004

### 附录















































