数

据

结

构

2021课程设计

校园导览系统

陈睿扬 王仕和 张晏宁

2019213686 2019213681 2019213684

2021年6月10日

目录

1. 课程设计内容

1.1.课程设计目标

1.2.课程设计目的

1.3.课程设计开发者

1. 软件概述及用户使用说明

2.1.实现功能

2.2.软件包结构与软件使用

1. 总体方案设计说明

3.1.软件开发工具及环境

3.2.总体结构

3.3.模块划分

1. 数据结构说明和数据字典

4.1.核心算法

4.2.数据结构说明

1. 各模块设计说明

（分模块描述）

1. 应用范例执行结果及测试情况说明
2. 评价和改进意见

1.课程设计内容

1.1.课程设计目标

设计校园导览系统，实现以下需求：

1.校园导览系统具备导航功能，能提供查询服务，内容包括当前时刻同学所处的地点，周围的服务设施信息。

2.校区、校园内建筑设施、道路具有基本的丰富性和复杂性。

3.实现导航时物理位置和逻辑位置的一对一对应，支持校区间导航，交通方式为可自行规划班次时刻表的定点班车。

4.实现《课程设计指导》所提到的4种导航策略，实现最短距离的导航、实现最短时间（拥挤度设置）的导航、实现多点途径最短导航、实现校区内交通工具的改变。

5.实现依据时间推进和可视化，支持调节时间倍率。

6.已建立日志文件

7.实现点击查询学生当前位置和周围服务设施，可通过从当前点出发更改终点的方式获取到周围服务设施的导航。

8.选做已完成内容：可更改目标和导航策略，系统按照当前位置做相应的操作；用图形绘制地图，并在地图上进行实时导航。

1.2.课程设计目的

1.巩固课堂讲授和书本知识，以理论联系实际的方法进行教和学的实践

2.提高分析和解决实际问题的能力

3.训练开发软件和书写软件文档的能力

1.3.课程设计开发者

|  |  |  |  |
| --- | --- | --- | --- |
| 学院 | 班级 | 姓名 | 学号 |
| 计算机学院 | 2019211301 | 陈睿扬 | 2019213686 |
| 计算机学院 | 2019211302 | 张晏宁 | 2019213684 |
| 计算机学院 | 2019211302 | 王仕和 | 2019213681 |

2.软件概述及用户使用说明

2.1.实现功能

1.普通寻路模式

实现点到点的寻路，支持如1.1中所述的各项选择，支持修改目的地。

2.选点寻路模式

实现选点需求和多点导航（旅行商问题等）

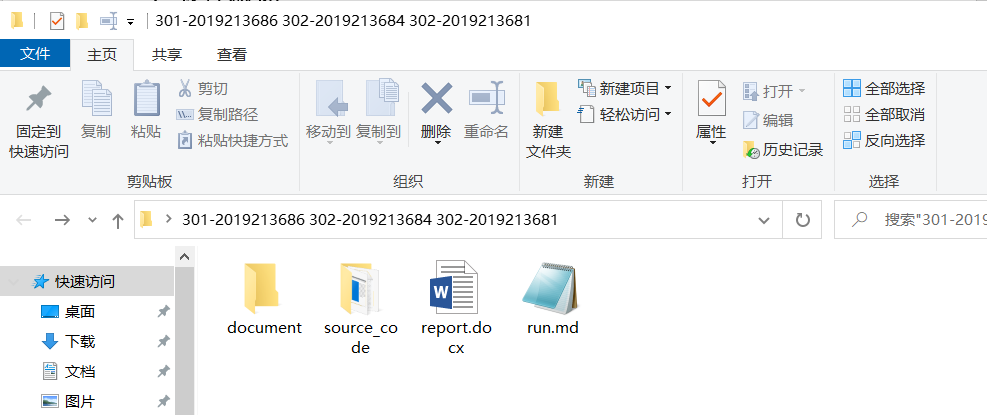
3.查询和可视化

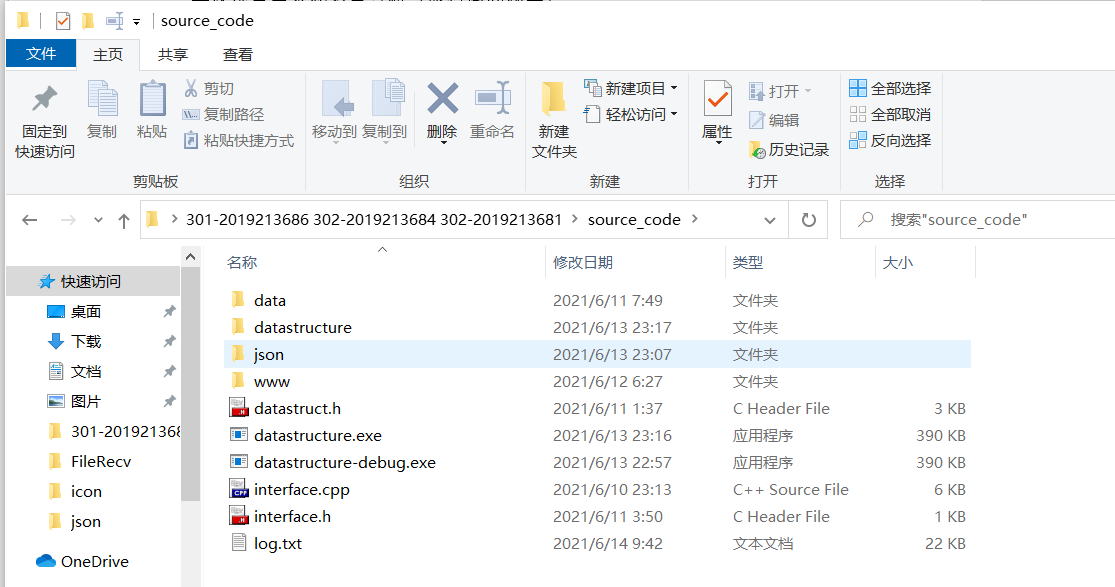
提供校区地图和寻路表现，支持点击地图进行操作。

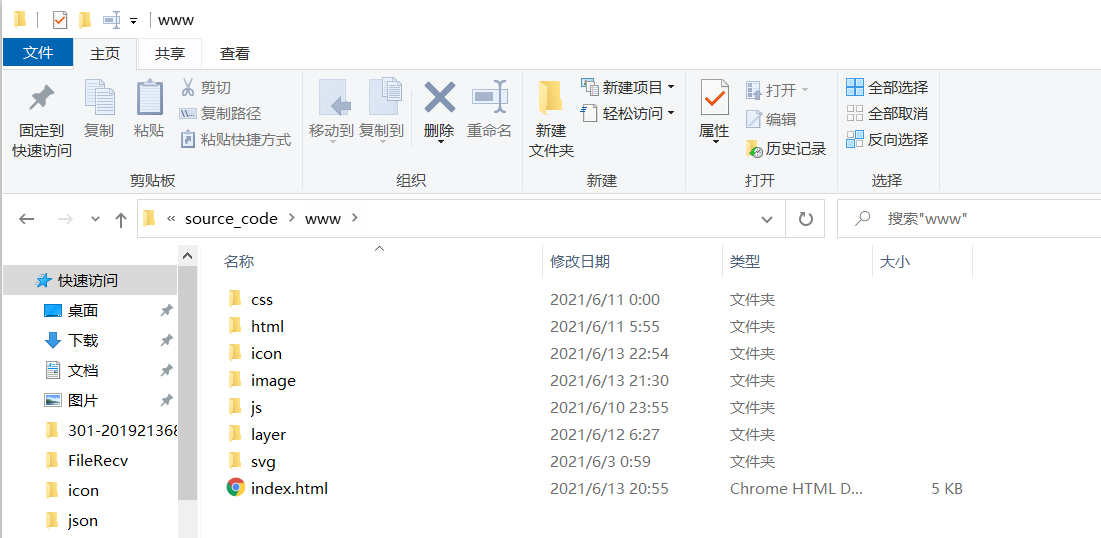
2.2.软件包结构与软件使用

如下所示，软件包含完整的三端架构，使用前需要运行自建服务器datastructure.exe,默认端口为127.0.0.1:8888,在此位置运行主文件index.html即可。由于其网页化的特性，需要在浏览器以html的方式运行。相关文件依赖按照原文件路径排列即可。

如界面图，左端为操作模板，右端为输出模板与可视化模板，查询、选点等功能可在地图上点击实现。本课程设计拥有较高程度的可视化，在页面打开后按照常规逻辑和页面提示操作即可。









3.总体方案设计说明

3.1. 软件开发工具及环境

* Windows 10系统（64位）
* [Microsoft](https://www.baidu.com/link?url=zzKtQvEGTIqKJTjd6rXHhTC_tZUpN-ozsqxgwA53EQDk1H9FrF4CodpLmQ3Ty5pEOKewvjs2TuuOjNZimRzWW-hgeYLnuls0tdoThQWpJv7&wd=&eqid=a72381df003f27140000000360c6a88f) Visual Studio 2019
* [Visual Studio Code](http://www.baidu.com/link?url=CnQ_7WdgoOVdU7KZgEQO-Jt8Xcu9b1iIRHyIBVgfyvc6NKJaewPW0yNrDC5a3Nfg)
* DEV-C++ IDE

3.2. 总体结构

系统由多种语言完成开发，大抵可分为三个部分。

1. 图形用户界面

使用HTML、CSS、JavaScript进行开发，独立书写的HTML、CSS构成了可视化效果良好的图形用户界面，echart.js提供了丰富的地图组件，使用JavaScript丰富的可交互性提供了良好的用户体验。能够实时显示旅行线路与状态，动态更换目的地位置。

2. 服务器部分

使用C++进行开发，采用了C++17语言标准，基于boost库的开源轻量级http服务器cinatra库为我们提供了高效的服务器实现，采用rapidjson可以高效的将前后端通信的文件编码为json格式进行传输。独立书写的各个api可供前端进行调用，为图形界面的输入和核心算法部分的输出构建了高效且合理的桥梁。

3. 核心算法部分

使用C++进行开发，使用堆优化的迪杰斯特拉最短路算法，将时间复杂度优化为，可使得计算时间在图规模增加时增长缓慢，为快速计算校园导览系统所需的路径提供了有力保障。

3.3. 模块划分

3.3.1 用户图像模块

用户图像模块主界面接收鼠标及键盘输入命令，分析该命令并调用其他模块，以时间为轴向前推进，并提供了时间倍率修改模块。

3.3.2 核心算法模块

通过C++语言，实现了高效的最短路算法，为其他模块提供了底层的计算功能，并使得程序时间复杂度降至最低。

3.3.3 服务器处理模块

通过网络编程，实现接收用户图像模块主界面传递的命令，调用、处理、分析核心算法模块的导航线路及规划。

3.3.4导航线路输出

根据服务器提供的数据，将导航信息可视化的输出到图形化界面上。

3.3.5. 建筑物和服务设施查询显示模块

可以通过点击当前位置，查询位置周围的建筑物和服务设施，也可以进一步计算距离服务设施的距离。

3.3.6. 日志文件处理模块

将开始规划与规划成功等日志记录在文件中，方便日后进行查询工作。

4. 数据结构说明和数据字典

4.1. 核心算法

核心算法部分使用堆优化的迪杰斯特拉最短路算法。我们知道，在使用迪杰斯特拉算法计算最短路时，每轮需要选出当下距离起点距离最短的点作为中间节点进行更新，若使用简单的搜索算法选出该最短距离点，总复杂度为。为了提高程序效率，我们使用“优先队列”数据结构来优化搜索过程。优先队列实质上为一个“堆”结构，具有查询、插入的性质。因此，可以将总复杂度优化为，大大提高了程序运行效率。若对于跨校区的路程，我们分别找出起始点到所在校区的上车点、终点到所在校区的上车点，然后根据发车时间、到达时间搜索找到最近的一辆定时班车或公交车，将该班车信息加入推荐路径。

4.2. 数据结构说明

：节点个数。

：路径条数。

struct Node {

    int nType;

    int nCampus, nId;

    std::string nName;

    Node() {}

    Node(int nType, int nCampus, int nId, std::string nName) :

        nType(nType), nCampus(nCampus), nId(nId), nName(nName) {}

};

vector<Node> pNode;

：节点集合，用于存储所有节点。

struct Edge {

    int eType;

    int eCampus;

    int eStart, eEnd;

    double eCrowd;

    double dis;

    Edge() {}

    Edge(int eType, int eCampus, int eStart, int eEnd, double eCrowd, double dis) :

        eType(eType), eCampus(eCampus), eStart(eStart), eEnd(eEnd), eCrowd(eCrowd), dis(dis) {}

};

vector<vector<Edge>> mymap;

：用于存储每条边的信息。其中：表示道路类型，分为通人、通自行车、通车三类；：表示该地点所属的校区；：道路连接的两端点；：道路拥挤度；：道路长度。

：存储图的数据结构，采用邻接表方式存储，在c++中使用vector进行存储。

struct Bus {

    int bType;

    int bStCamp, bStId;

    int bEnCamp, bEnId;

    double bStartTime;

    double dis;

    Bus() {}

    Bus(int bType, int bStCamp, int bStId, int bEnCamp, int bEnId, double bStartTime, double dis) :

        bType(bType), bStCamp(bStCamp), bStId(bStId), bEnCamp(bEnCamp), bEnId(bEnId), bStartTime(bStartTime), dis(dis) {}

};

vector<Bus> busTable;

：用于记录班车或公交车信息。其中：表示车辆的种类，可以是定时班车或公交车；：表示班车的起始校区和编号、终点校区和编号；：表示班车的发车时间；：表示班车需要行驶的距离。

：存储公交车信息的数据结构，采用简单的数组方式储存，在c++中使用vector进行存储。

：在最短路算法中记录一个点是否访问过。

：在最短路算法中记录一个点的最短路方案，仅记录到达该点的前一条边。

：记录最短距离。

：用于生成起点到终点的最短路，由遍历中某些节点得到

：用于堆优化的优先队列。队列中每个节点有两个字段，分别代表点编号与距离。

：用于存储每个点的信息。其中：表示节点类型，可分为普通点、服务设施、不存在三类；：表示该地点所属的校区；：表示地点的编号；：地点名称。

：用于记录生成的最短路的各类信息。其中：该路段类型，与类似；：分别表示该路段的起始校区和编号、终点校区和编号；：到达该路段的终点端点时已消耗的时间与通过的路程；：该路段连接的下一条路段。

：用于记录用户提出的行程要求。其中：表示用户的策略，大致可分为时间优先与距离优先；：表示用户的交通工具，可以步行或骑车，与上一字段组合可以构成最终的优先方案；：用户名；：表示用户的起始校区和编号、终点校区和编号；：表示用户的出发时间；：记录经过最短路算法后得到的路径，分别表示道路的段数与首条道路。

若干常量的定义

：分别为沙河校区、西土城校区上车点编号。

：分别表示步行、骑车、公交车与定时班车的行驶速度。

：表示点的类别，分别为：不存在、普通点、服务设施。

：表示交通工具的类别，分别为：步行、骑车、坐车。

：表示优先策略的类别，分别为：距离优先、时间优先。

：表示车辆的种类，分别为：定时班车、公交车。

5. 各模块设计说明

5.1 用户图像模块

通过html语言在网页端进行布局与样式设计，具体所用技术参照5.7 前端模块具体设计。

5.2 核心算法模块

高效的核心算法设计，参照4.1核心算法

5.3 服务器处理模块

通过网络通信完成，参照5.8后端模块具体设计

5.4导航线路输出

设计为在网页上创建导航结果输出栏，根据服务器端给出的数据进行输出，呈现为前端的布局样式。

5.5. 建筑物和服务设施查询显示模块

设计为点击图标产生的tooltip事件，通过formatter回调实现内容个性化显示。

5.6. 日志文件处理模块

会在文件目录下生成日志文件log.txt,具体内容设计为由后端导出每次运行的状态，并进行存储

5.7．前端模块具体设计  
 课程设计使用HTML + CSS + JavaScript作为前端架构，以实现数据可视化。图像化界面由两个主colomn构成，分别进行页面操作接入与地图构成，以及结果的输出。地图元件使用echarts和svg矢量图像进行模拟，实现包含提示窗口查询、路径可视化、地图结构可视化与中途操作可视化等功能。样式采用bootstrap.min.css与index.css联合设计，实现较为合理的布局展现。前端逻辑构成与前后端数据交互通过js文件组和ajax方法进行通信实现，以此完成完整的页面渲染与数据加载，于此同时，设计采用一定量的图形优化以提高使用者的使用体验。本设计前端完整地实现了需求的可操作化，并且在样式美化与操作简化方面进行了严谨的考虑。

5.8.后端模块具体设计

使用一个c++的http服务器框架，实现了算法部分与前端部分的通信，main.cpp中定义了各种api的接口即json文件格式，通过cinatra实现了http服务器，rapidjson实现了c++处理json格式文件，方便快捷的写入json格式，以供前端调用。其中datastruct.h定义了数据结构，datastruct.cpp抽象出一个可供调用的接口，使得可以计算出各种模式下的最短路。interface.h和interface.cpp实现了各种api调用的函数，使得模块化合理，可拓展性强。

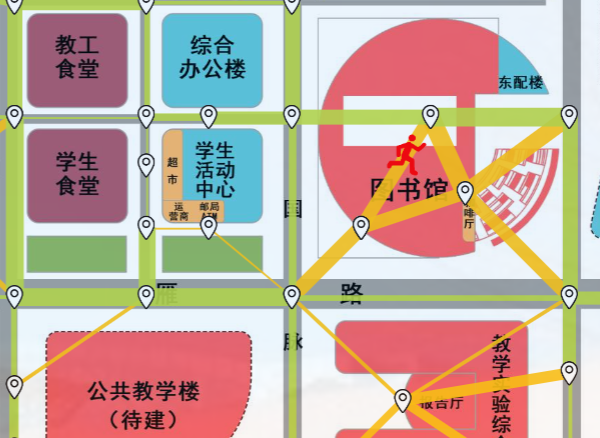
5.9.算法模块具体设计

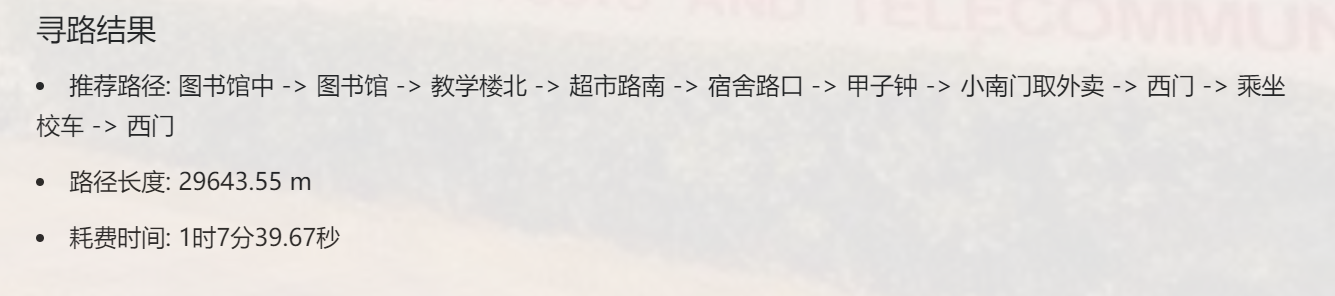
核心算法部分使用堆优化的迪杰斯特拉最短路算法。我们知道，在使用迪杰斯特拉算法计算最短路时，每轮需要选出当下距离起点距离最短的点作为中间节点进行更新，若使用简单的搜索算法选出该最短距离点，总复杂度为。为了提高程序效率，我们使用“优先队列”数据结构来优化搜索过程。优先队列实质上为一个“堆”结构，具有查询、插入的性质。因此，可以将总复杂度优化为，大大提高了程序运行效率。若对于跨校区的路程，我们分别找出起始点到所在校区的上车点、终点到所在校区的上车点，然后根据发车时间、到达时间搜索找到最近的一辆定时班车或公交车，将该班车信息加入推荐路径。

6. 应用范例执行结果及测试情况说明

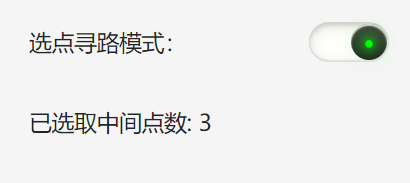
程序拥有自刷新功能，可支持连续多次运行，同时也支持网页端刷新。

以普通寻路模式下沙河图书馆到西土城西门为范例1，执行结果如下。测试结果符合预期，可正常得出寻路结果（路径长度、时间等），可视化可正确运行。

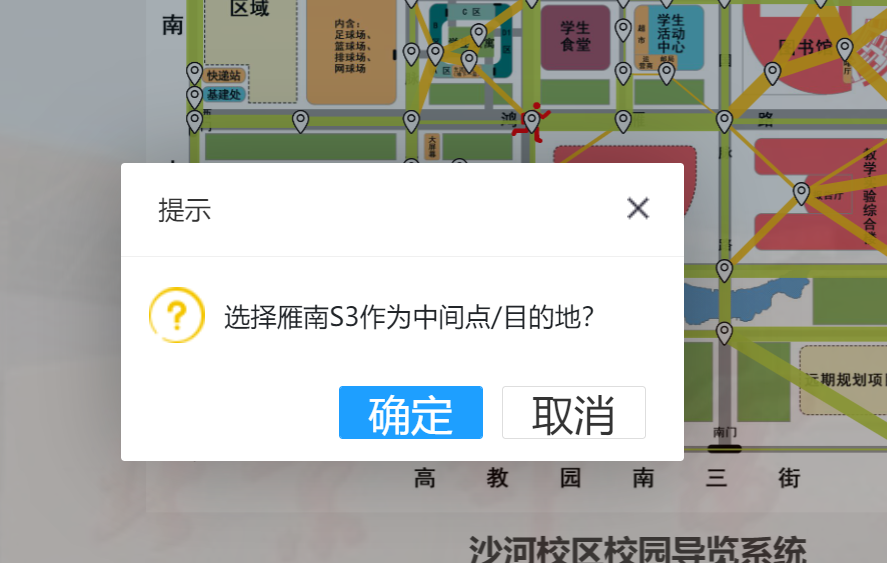
 

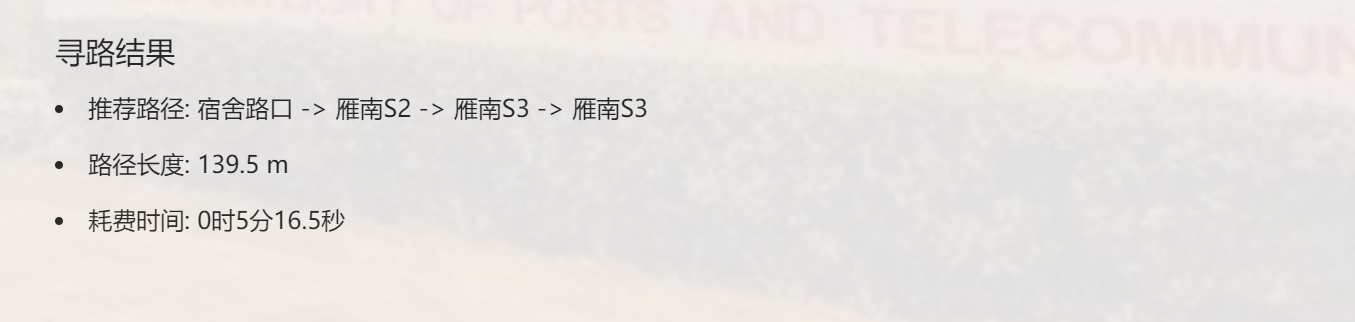


以选点寻路模式下宿舍路口为出发点、S2、S3、S4为中间点为范例2，执行结果符合预期。

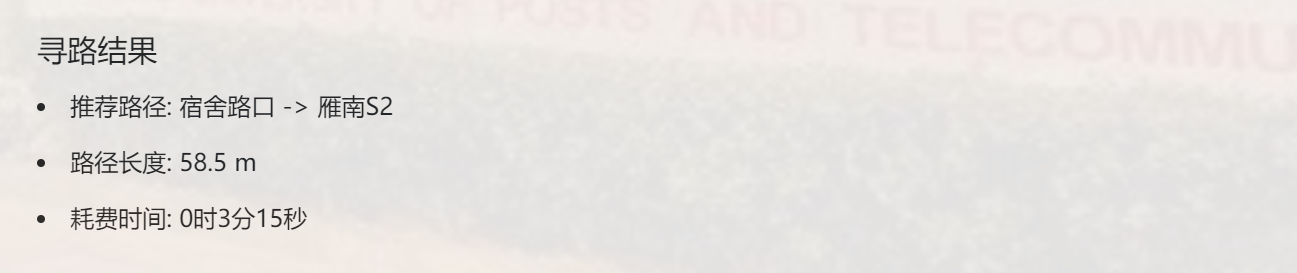


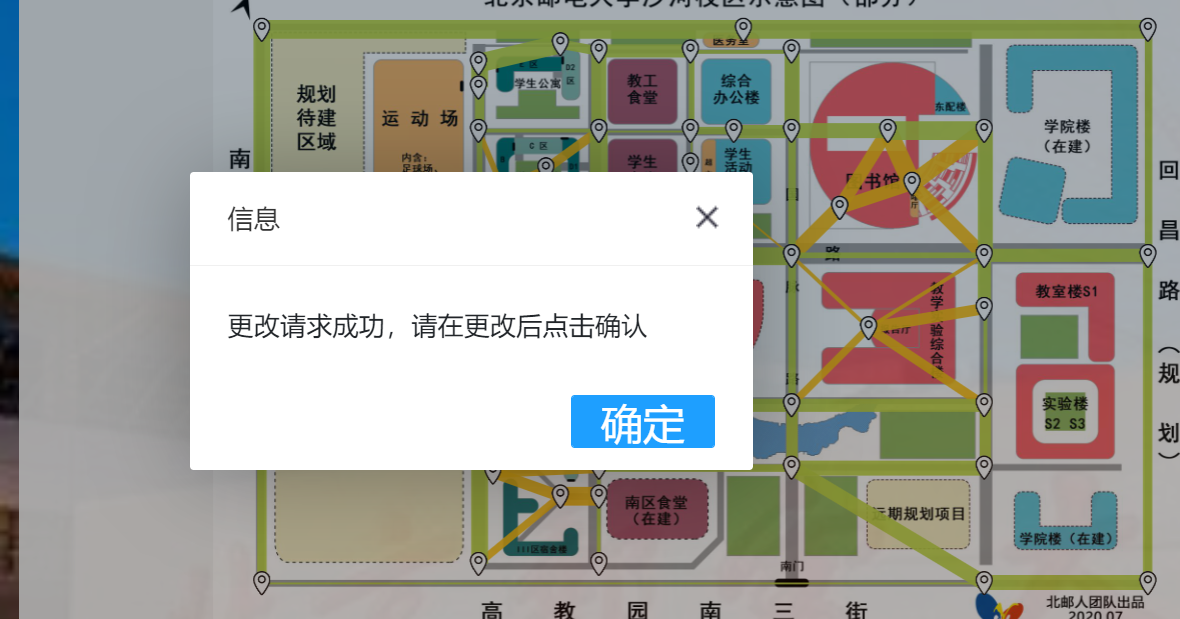


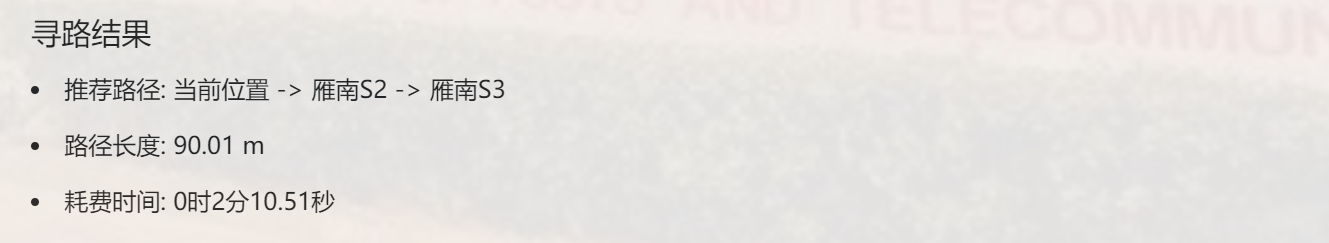




以普通寻路模式下宿舍路口为出发点、S2为中间点，执行更改终点为S3，作为为范例3，执行结果符合预期。









7.评价和改进意见

评价：本课程设计基本充分的实现了实验指导所提出的需求功能，在表现形式，逻辑结构和算法效率方面都达到了较好的水平。

改进意见：在前端架构方面，可增加成熟框架的使用以减少代码的编写量，改善总体表现形式；在后端服务器方面，可设计更加稳定严谨的通讯协议，优化算法返回结果的处理，定义更加具有可复用性和可延展性的接口；在算法设计方面，可以增加算法前沿技术进行特定情况下的优化，采用更灵活的设计以提高算法的可拓展性。同时，可增加选做部分的楼内导航功能作为附加实现。