

Project4

中山大学校园导游咨询

班级：教务 4 班 / 行政 8 班

姓名：钟文谦 学号：16340309

姓名：周伟标 学号：16340310

姓名：周泽昊 学号：16340312

姓名：章恒 学号：16340298

【问题描述】

1. 从中山大学东校区的平面图中选取有代表性景点(20-30 个), 抽象成一个无向带权图。以图中顶点表示景点, 边上的权值表示两点之间距离。
2. 本程序的目的是为用户提供路径咨询。根据用户指定的起点和终点输出相应的路径, 或根据用户指定的景点输出景点的信息。

【任务要求】

1. 从中山大学东校区的平面图中选取有代表性景点(20-30 个), 抽象成一个无向带权图。以图中顶点表示景点, 存放景点名称、代号、简介等信息; 以边表示路径, 存放路径长度等信息。
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问路查询, 即查询两个景点之间的一条最短的路径。

【此次 project 做出的调整】

经过我们组内成员的讨论, 针对我们目前团队的优劣, 我们做出了较大的调整:

1. 首先, 是对于 project 的分工以及要做什么做出了调整, 以前大致都是一个人负责设计 C++ 程序的大致结构并完成一半的代码, 一个人根据需求完成另外一半代码, 然后一个人负责测试, 经过思考, 我们觉得经过这一个学期在 WEB 方面较为深入的学习, 根据我们现在的能力, 我们可以不再局限于黑框程序, 我们决定一个人完成基础的 C++ 部分, 一个人完成 WEB 的部分, 另一个人继

续负责测试。

2. 其次呢，这一次我们之前小组经过内部讨论，招来一个敢于踊跃上去做展示的人(我们之前几次都没有敢于上去分享，即使我们自我感觉做的还可以)，我们觉得我们团队就缺这一个带我们一起上去分享的人。

【数据结构与算法】

C++部分：

1. 由于本实验注明是 20~30 地点所以选择矩阵来存储对空间的占用不会很大，而且操作很方便，对于地点名称以及描述，我们将它存储在一个数组中（顺序对应着矩阵的顺序），如下图：

```
struct place{
    string name;
    static int count;
    int id;
    string description;
    place():name(""),description(""){}
    place(string n,string d):name(n),id(count++),description(d){}
};

class graph {
private:
    int Manmatrix[500][500];
    int Carmatrix[500][500];
    place list[500];
    int num;
```

2. 对于人行、车行路线的增删操作以及地点的增加很简单这里不做过多赘述，这里提一下地点的删除，地点删除时，会直接将数组中最后一个地点移动到当前位置，这样可以很有效的简化操作
3. 最小路径算法，采用迪杰斯特算法，通过一个 `vector<int>`的数组来存储这个点到其他各个点的最短路径（一开始每个与起始点有连接的点会存储起始点，每当路径改变时，将改点的路径改变为与前

面那个点的路径，并在其中路径中加入前面那个点)，这样输出时，只需要输出对应的路径即可

4. UI 部分，实际上整个程序并不复杂，但是 UI 却是最麻烦的地方，这里承包了各种异常处理（并没有什么可讲的...）

WEB 端部分：

1. 首先从最基础的部分说起吧， 初期，经过考虑，为了方便数据持久化以及数据的可管理，我决定把地点数据存入数据库中进行管理，所以需要设计一种方便读取以及存储的数据结构。初步设计后，我把图的数据分成景点(即点)以及线路(即点之间的连线)进行储存，即以下的数据结构：

(meta 存放的是地点的各种小信息，比如被赞数等，point、comments 则是地点评分、评论)

```
var placeSchema = new Schema({
  name: { type: String, required: true },
  value: { type: String, required: true },
  introduce: { type: String, required: true },
  x: { type: Number, required: true },
  y: { type: Number, required: true },
  meta: { type: Schema.Types.Mixed },
  point: { type: Number, required: true },
  comments: [ Schema.Types.Mixed ]
});
```

(source 为线路的起点，target 为终点，value 为路径长度，type 用于线路类型的区分(0:人行道，1:车行道))

```
var linkSchema = new Schema({
  source: { type: String, required: true },
  target: { type: String, required: true },
  value: { type: Number, required: true },
  type: { type: Number, required: true }
});
```

1. 确定存入数据库的点和线的数据结构，剩下的就是图的构成了，我使用的是邻接矩阵对图进行处理的，在用户打开网页后，会通过 AJAX 请求，向服务端获取地点数据以及路线数据，然后通过处理，转换成图。转换成图的部分算法较为简单，跟老师教的图的输入类似，这里就不进行赘述了。
2. 在完成结构的设计之后，最关键的就是最短路径查找算法了，在图的结构完成后，其实最短路径的查找算法较为简单，使用基本的 dijkstra 算法，再加上对于一个数据数组的维护，把每一个都置为起点的编号，然后在某个点的最短路径刷新时，把其编号对应数组内的数值改为他前一个点的编号，在完成整体的搜索后，通过一个简单的类似并查集搜索原点的算法就可以把相应的路径找出来，即以下的代码：（这样就可以较为简单地得到 Path 的整个路径）

```
var path = [], pathString = "";
while (temp[v] != v) {
    path.unshift(v);
    v = temp[v];
}
path.unshift(v);
for (var i = 0; i < path.length; i++) {
    if (i != path.length - 1)
        pathString += placeName[path[i]] + "->";
    else
        pathString += placeName[path[i]];
}
```


3. 其次就是地点路线的增删改查了，有了对于数据库操作的基本知识后，其实数据的增删改查的难度不大，关键就是数据去重以及数据删除、更改时对于其他受影响数据的处理，毕竟地点和线路之间是

有相互联系的。算法方面其实没什么难度，只要把情况考虑完善就可以了

【测试数据、结果与分析】

C++部分

文件读入

 选择D:\桌面\导游 (1)\导游.exe

你有已有地图进行输入吗(0: 无, 1: 有)? 1
请输入文件名:
test
成功导入!

添加地点

-----导游-----
1 - 添加地点
2 - 添加或修改步行路径
3 - 添加或修改车行路径
4 - 删除地点
5 - 删除步行路径
6 - 删除车行路径
7 - 查询地点信息
8 - 寻找最短步行路径
9 - 寻找最短车行路径
10 - 输出所有地点
11 - 读取已有地图
12 - 存储地图
13 - 离开

请输入你的选择: 1
请输入地点名称:
数据院
请输入简短的介绍:
学院
成功添加!

(以下省略菜单)

添加路径

请输入你的选择: 2
请输入地点1名称:
北校门
请输入地点2名称:
数据院
请输入距离 (正整数)
200
成功添加!

删除地点

请输入你的选择：4
请输入地点名称：
公交车站
成功删除

删除路径

请输入你的选择：5
请输入地点1名称：
北校门
请输入地点2名称：
数据院
成功删除

查询地点信息

请输入你的选择：7
请输入地点名称：
北校门

名称：北校门
编号：6
描述：北门
临近(步行)地区：饭堂 距离：500
南校门 距离：1500
数据院 距离：200
临近(车行)地区：地铁站 距离：500
南校门 距离：1000

寻找最短路径

请输入你的选择：8
请输入地点1名称：
宿舍
请输入地点2名称：
实验楼
宿舍--100->公教楼--200->实验楼
总长300

若无路径输出无路可走

请输入你的选择：8
请输入地点1名称：
宿舍
请输入地点2名称：
地铁站
无路可走！

输出所有地点

请输入你的选择：10
宿舍
公教楼
饭堂
实验楼
地铁站
数据院
北校门
南校门

储存

请输入你的选择：12
请输入文件名：
test1
成功存储！

WEB 端（管理端）

Sysu Map Manager

三 工具栏

添加地点

添加线路

地点以及路线列表

Simple Map of Sysu

Car FootWay All

地点列表

地点名称	评分	被赞数	评论数	管理
北门牌坊	5	5	1	查看 编辑 删除
行政楼	0	0	0	查看 编辑 删除
公共教学楼	0	0	0	查看 编辑 删除
图书馆	0	0	0	查看 编辑 删除

路线列表

起点	终点	类型	管理
北门牌坊	行政楼	人行道	编辑 删除
北门牌坊	公共教学楼	人行道	编辑 删除
行政楼	公共教学楼	人行道	编辑 删除

添加地点

地点详情

添加地点

地点名称

公交车站

地点介绍

坐公交车

X坐标

10

Y坐标

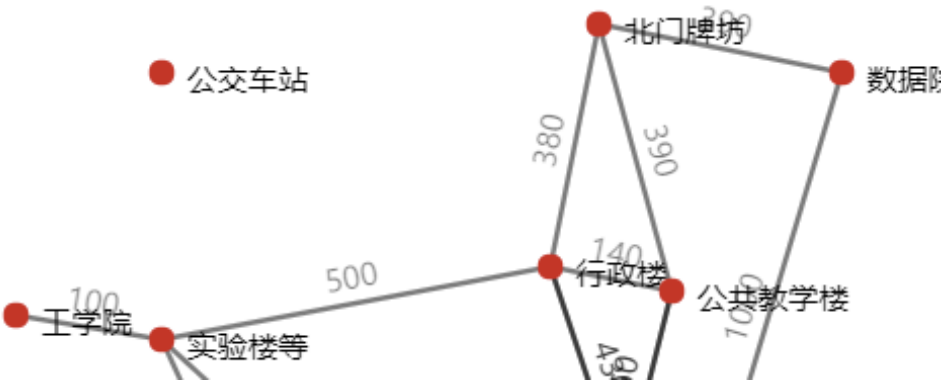
10

提交

关闭

确认

地点以及路线列表



查看编辑删除地点

公交车站	0	0	0	<div>查看编辑删除</div>
------	---	---	---	-------------------

查看（可以点赞评分评价）

地点详情

地点名称: 公交车站

地点介绍: 坐公交车

被赞:0

★ 评分:0

地点评论

还没有评论哦！快点去留下你的心声吧！

添加评论

邮箱

请输入邮箱...

内容

请输入内容...

请选择对该地点的的评分

0

提交

关闭

确认

编辑修改信息

地点详情

地点信息修改

地点名称

公交车站

地点介绍

坐公交车

X坐标

10

Y坐标

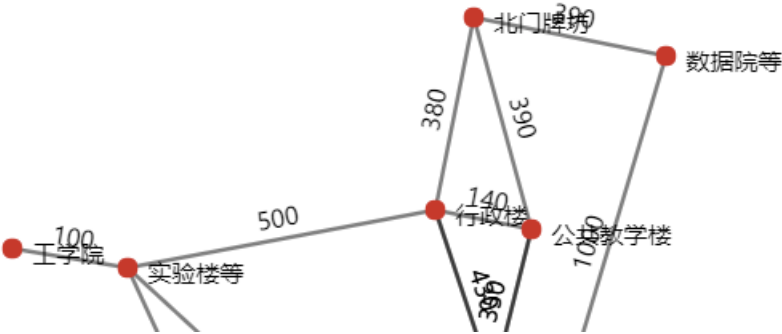
10

提交

关闭

确认

删除



添加路径

添加路线

路线添加

起点

北门牌坊

终点

大学城北地铁站

类型

车行道

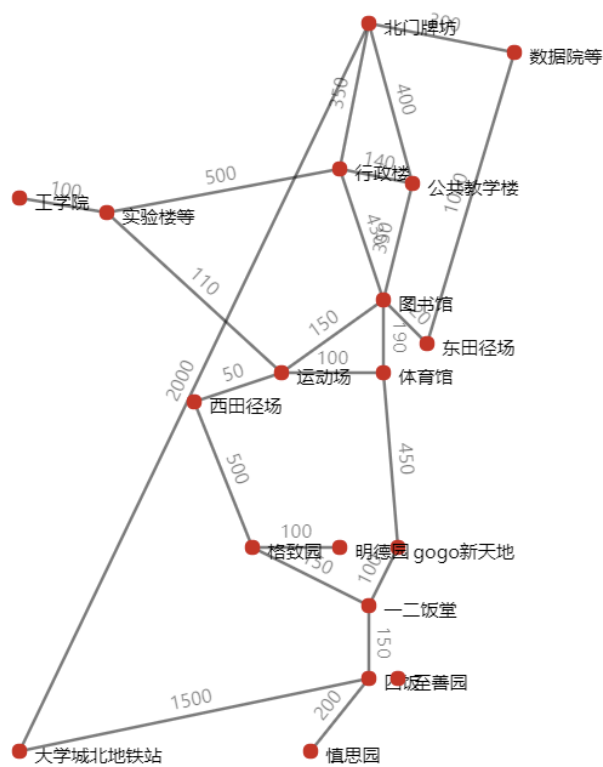
长度

2000

提交

关闭

确认



Simple Map of Sysu

- Car
- FootWay
- All

北门牌坊	大学城北地铁站	车行道	编辑	删除
------	---------	-----	----	----

编辑路径

修改路线信息

路线信息修改

起点

北门牌坊

终点

大学城北地铁站

类型

车行道

长度

2000

提交

关闭

确认

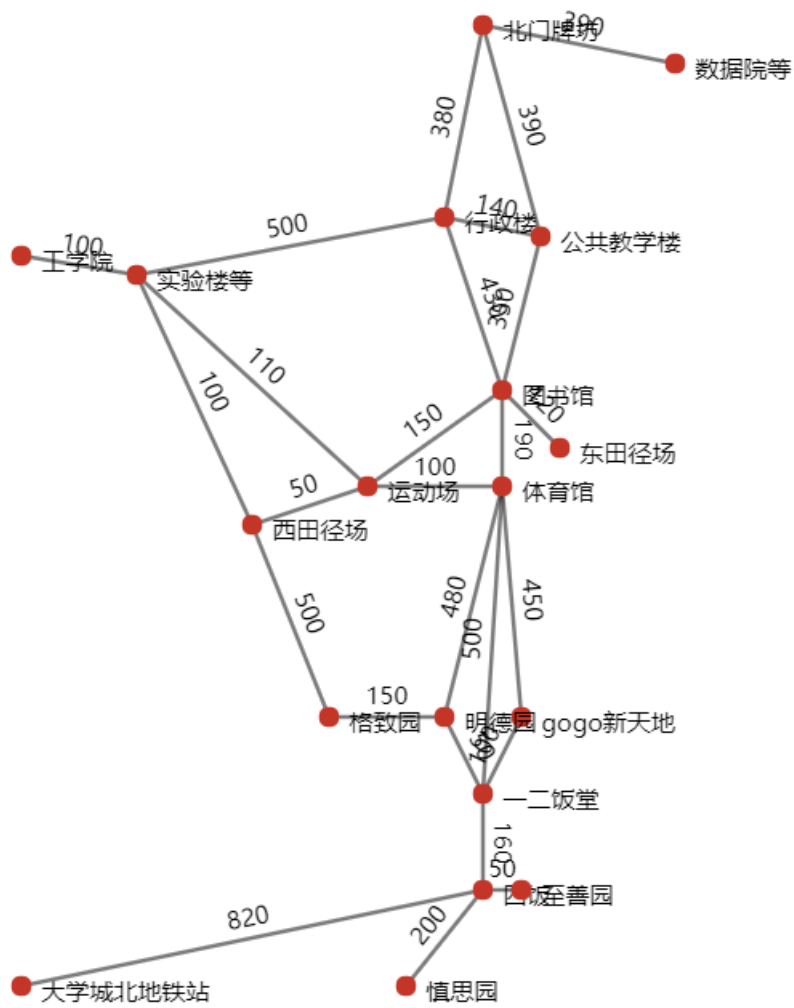
删除路径



FootWay

All

人行路径



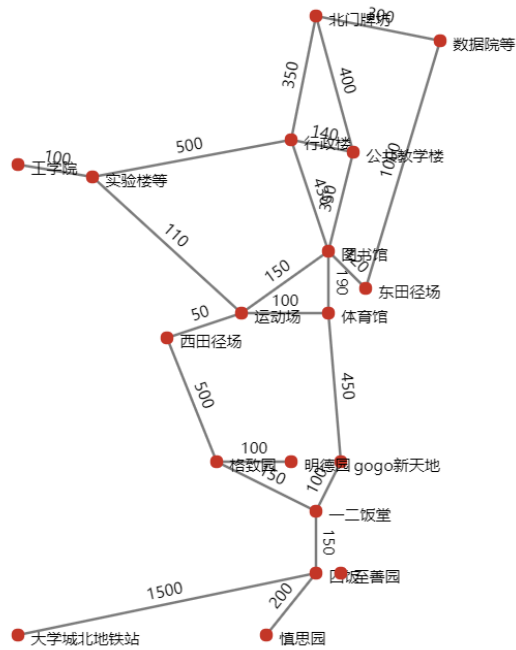
Simple Map of Sysu

Car

FootWav

All

车行路径



Simple Map of Sysu

[Car](#) [FootWay](#) [All](#)

WEB 端（客户端）

← → ↻

192.168.199.194:3000

Sysu Map

Home

Blog

Simple Map of Sysu

[Car](#) [FootWay](#) [All](#)

路径查询小助手

起始地

北门牌坊

目的地

北门牌坊

确认

查询路径

路径查询小助手

起始地

北门牌坊

目的地

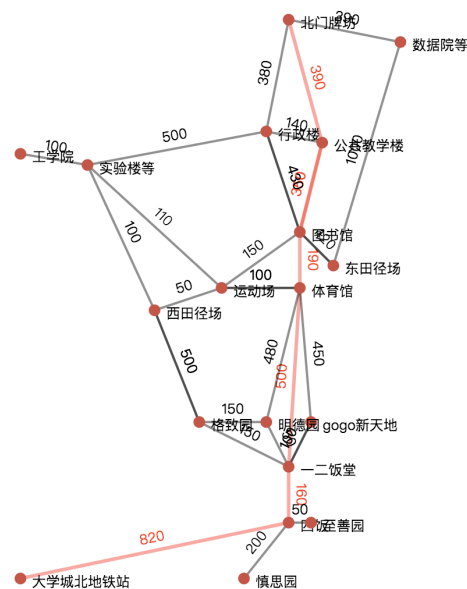
至善园

确认

Path: 北门牌坊->公共教学楼->图书馆->体育馆->一二饭堂->四饭->至善园

Length: 1640米

分享你的路线



Simple Map of Sysu

路径查询小助手

起始地

北门牌坊

目的地

大学城北站

确认

Path: 北门牌坊->公共教学楼->图书馆->体育馆->一二饭堂->四饭->大学城北站

Length: 3090米

分享你的路线

点击分享你的路线

路途分享小博客

非常美妙的一次经历

作者: 1316628630@qq.com Path: 北门牌坊->行政楼->图书馆->体育馆->东田径场

分享内容:

这个软件很好用, 清晰地展现了路线以及最快的路

❤ 被赞:28 💬 评论数:2

这个路线还可以

作者: 623@qq.co Path: 北门牌坊->公共教学楼->图书馆->体育馆->明德园

分享内容:

还彳亍吧!

❤ 被赞:1 💬 评论数:0

很方便

作者: test@qq.com Path: 北门牌坊->公共教学楼->图书馆->体育馆->一二饭堂->四饭->大学城北地铁站

分享内容:

之前来大学城找不到地铁站, 有了这个, 很方便就可以找到一条迅速的路径

❤ 被赞:0 💬 评论数:0

对于博客, 我们同样添加了点赞以及评论功能, 这里就不一一赘述了

【分工、贡献%、评分】

钟文谦: 完成 WEB 端 app, 30%, 90

周泽昊: 完成 c++ 版本, 30%, 90

周伟标: 负责数据收集、测试, 20%, 90

章恒: 上台分享, 部分 idea 提出, 20% , 90

【项目总结】

经过这次项目,

1. 我们小组对于图的节点的插入、删除, 路线的插入、删除, 以及最短路径的算法有了更多的理解, 并且得以把课上学习到的东西以及实验做的东西应用到实际的应用上, 做出一个实际的产品。
2. 我们对于团队交流的作用有了更多的体会, 经过讨论, 我们做出决策的变化, 完善了我们小组的缺点, 引入了一个新成员, 然后

基于能力的进步，对于分工以及产品做出适当的调整，感觉整个团队的运作更加合理了。

3. 这一次次项目都是把我们数据结构学习到的数据结构以及算法应用到我们生活的实际问题，让我们体会到了老师的良苦用心，希望我们学会把程序、把我们所学应用到我们的实际生活中，让我们在今后的生活中，更加地注意去发现生活中的各种实际问题，或者各种有趣的想法，然后通过编程去解决。通过一次次在生活的实际问题中应用我们平时学到的各方面的知识，进而使我们更好地掌握课程的知识，体会到课程的实际意义，更加努力，带着兴趣去学习我们的课程。

【程序清单】

c++版本，放置在 `code/c++` 文件夹下

WEB (nodejs 为服务端) 版本，放置在 `code/map-guide-in-SYSU` 文件夹下，如果想直接体验可以直接访问 `zhongwqblog.xyz:8000`，不过由于服务器不在国内，速度不稳定，望见谅。

注：进入管理员端，直接访问 `localhost:8000/admin`

(`zhongwqblog.xyz:8000/admin` 同理)，然后注册一个账号，登陆后可以管理地图的信息了。