

数据库大作业实验报告

计 63

孙亚男

2016011285

一、题目描述

互联网拼车情景设定:

- 给定 10 万出租车
 - 当时经纬度
 - 乘客上限为 4 个
 - 车上乘客终点
- 给定 1 个查询——乘客当前位置
 - 找到最合适的 5 个车（不超过上限）
 - 出租车绕路不超过 10km
 - 出租车离乘客不超过 10km

其中绕路距离定义如下：

假定当前车上有 k 个乘客，依次送达目的地所需路程为 $D1$ ，出租车接乘客上车所需路程为 $D2$ ，接到乘客后将所有 $k+1$ 个乘客送达目的地所需路程为 $D3$ ；待接乘客位置到自己目的地的最短路距离为 $D4$ 。则车上乘客绕路距离为 $D2+D3-D1$ ，待接乘客绕路为 $D3-D4$ 。需要上述两个距离不超过 10km。

二、算法描述

首先我使用作业要求中推荐的 github 上 TsinghuaDatabaseGroup 的[GTree 仓库](https://github.com/TsinghuaDatabaseGroup/GTree)中的 GPTree。我调用了该文件中计算两个路网节点间最短距离的函数，以及构造完整路径的 api。在这个数据结构的支持下，我实现了我的算法：

- 1、数据计算 预先建立好 GPTree 并保存，预先计算初始状态下出租车将车上乘客送至目的地的距离。
- 2、乘客查询
 - a) 将保存好的数据载入
 - b) 从乘客出发路网节点开始沿路网进行宽度优先搜索
 - c) 对于可能满足要求的出租车，枚举送客顺序并调用 api 得到所需距离
 - d) 若满足条件，则加入搜索结果，否则继续搜索
 - e) 得到满足条件数量的出租车后，搜索停止
- 3、完善路径 调用 GPTree 的 api 对每个结果出租车计算送乘客的顺序和路线。

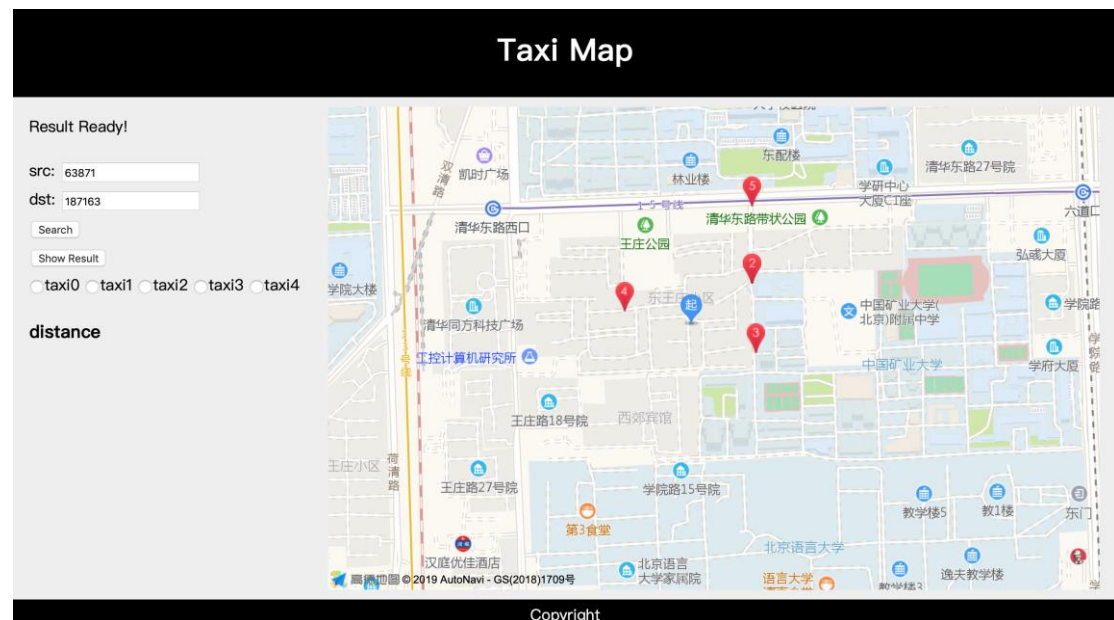
在宽搜的过程中，通过宽搜到的节点与源点的距离 D_2 进行剪枝，这样搜索速度提高，可以保证在很短的时间内枚举完所有可能的节点或者找到 5 个满足要求的出租车。

三、实验测试流程

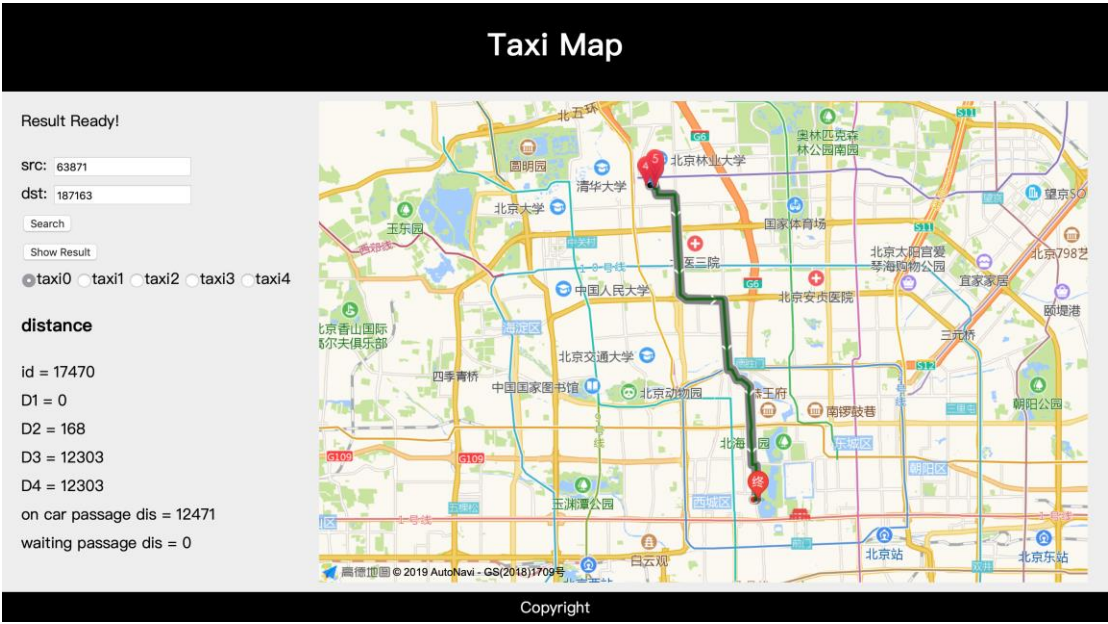
- 1、在根目录下运行 make data ，得到需要预先准备的数据。

- 2、在根目录下运行 `make run` ， 启动服务器端。
- 3、打开 `/result/map.html` 文件，运行拼车程序 UI。
- 4、待提示信息输出 “Map Ready !”后，输入出发路网节点编号和目的路网节点编号， 点击“Search”。
- 5、待提示信息输出 “Result Ready !”后， 点击“Show Result”。
- 6、点击单选框即可切换出租车查看相关距离、路径、乘客送达顺序等信息。

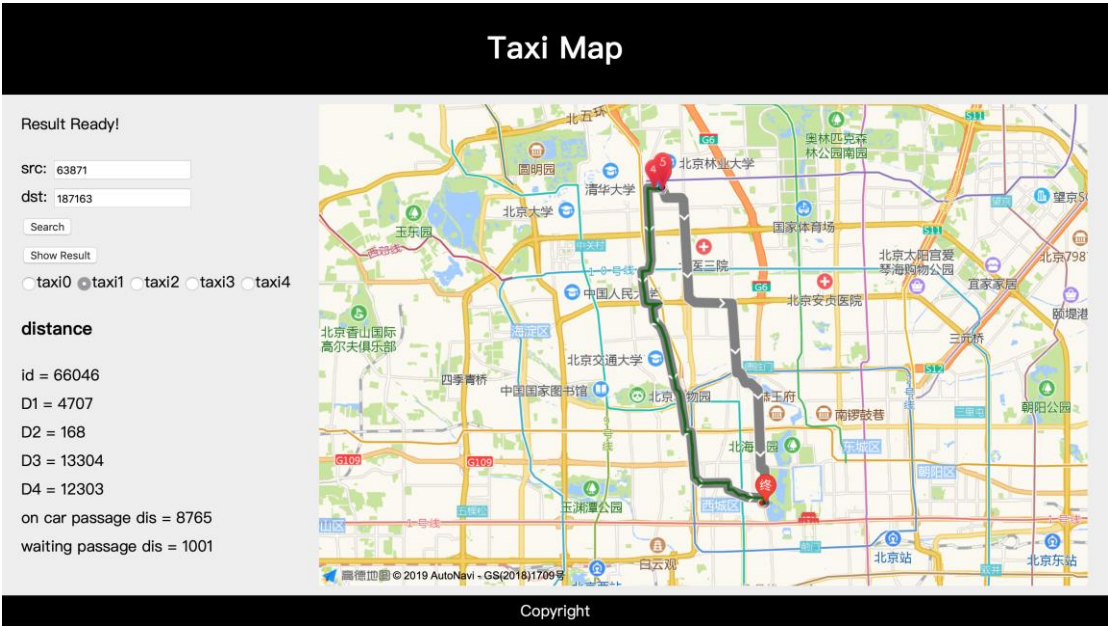
以检查测例为例：



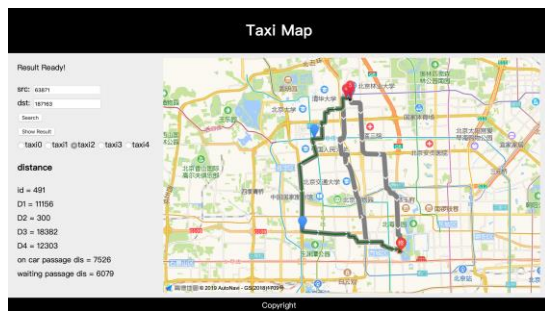
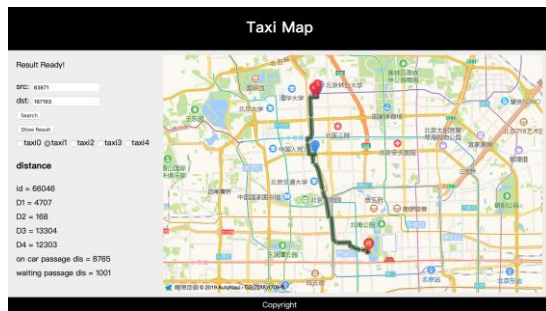
得到五辆出租车的结果，并在地图上标注（1 号和 2 号处在同一路口，所以图标重合）。



显示当前出租车相关信息。



可通过选中不同单选框查看不同出租车的信息。



若当前出租车上乘客，则通过蓝色点标注车上乘客下车位置。



可实现动画巡航，显示出租车行驶路径。

附：得到的 5 辆出租车 id 17470 66046 491 1605 46965

四、实验总结

本次实验考察对空间搜索相关知识的应用，加深了我对课堂知识的理解。

由于自己尝试写寻路算法没有成功，本次实验我调用了已有的库。但是在安装上碰到了很大的麻烦，被迫无奈借了一台电脑完成了本次实验。

界面实现上，我最先想到的是用 QT，因为与所调用的库都是 C++ 编写，更好对接。但受玄学 bug 困扰，不得不放弃 QT 另寻它法。

由于 C++ 与 js 的对接较为困难, 我最终选择通过 python 将二者相连, python 通过网络接口与 js 实现数据互通, python 通过 pipe 与 C++ 实现数据互通。这种方法相对较为复杂, 涉及到大量的数据传递。

地图显示我使用了第三方资源。最先使用的百度地图 api 由于坐标不同, 且百度地图的坐标转换接口十分难用 (一次只能转换 10 个点), 所以最终选用高德地图 api 进行地图显示。

实验最终实现了除自己实现寻路算法这一附加功能之外的所有的功能。感谢老师和助教的指导!