

数据库大作业实验报告

匡正非 2015011273 计52

1. 实验要求

本次实验要求完成一个在线的出租车查询系统，能够根据用户所在的位置自动匹配并给出合适的出租车供其选择。系统需要每次以一个坐标（或一个路图中的节点）作为输入，在5-10秒内提供不超过5个可选的出租车，且要求满足一下两个条件：

- 用户在搭乘该出租车后的绕路距离不超过10KM
- 该出租车上的所有用户的绕路距离不能超过10KM

两者的绕路距离计算方法为：假设当前车不接该乘客时运送已有的所有乘客所需路程为D1，接乘客所需的路程为D2，接完该乘客之后送完所有乘客所需的路程为D3，而乘客本人从起点到终点所需的距离为D4，那么，用户绕路的距离为D3-D4，其它用户的绕路距离为D2+D3-D1。

上述的距离，可以通过欧式距离和路网的最短路两种方式进行计算。除了完成基本的查询功能以外，还可以进一步提供UI、行进路线、以及送达乘客顺序等额外功能。

2. 完成情况

- 采用路网计算距离 已完成
- 在5-10秒内完成查询 已完成，最迟2秒之内相应
- 返回满足绕路距离的车辆 已完成，输出了所有满足条件的车辆，并对其按照两个绕路距离的和进行了排序
- 供用户使用的UI 已完成所有
- 使用百度地图API显示位置 已完成
- 显示乘客送达顺序 已完成

3. 程序运行的方法

1. 下载metis库。
2. 将三个输入文件放入根目录下的data文件夹下。
3. 直接在根目录下运行run.sh。如果运行失败，请尝试以sudo模式运行。
4. 第一次运行由于需要建立GTree等结构，因此需要较长时间完成。之后的运行会相对较快。

4. 算法设计

由于本人实现的最终方案是使用django完成web端，因此代码共分为前端后端两个部分。下面对两者的设计分别进行介绍：

后端查询

作为数据库专题训练的大作业而言，后端部分的查询算法实质上才是此次实验的重点。正因如此，这一部分的算法设计也是需要重点介绍的部分。首先，由于本人采用的是路网数据结构，因此第一步需要完成的便是在路网中关于最短路的计算。由于系统要求在线查询任意两点之间的最短路，无法通过传统的最短路算法进行求解，因此需要维护较为高级的路图结构来加速查询。和大多数人一样，本人最终采用的是由清华大学数据库组提供的GTree库，利用此库提供的GTree可以在很快的速度内计算出任意两点之间的最短路。

本人最终的实现在有了这一结构的支持下并不复杂：对于每次查询暴力枚举附近所有的出租车，然后通过枚举运送乘客的顺序分别计算载客前和载客后所需的运送总距离，从而得出该车的绕路距离，判断其是否满足要求。然而，这样的算法所需的时间较长，并不能满足5-10秒的要求，故又进行一些优化：

- 直接删掉了距离用户10KM以上的所有车辆。
- 直接删掉了已满的车辆。
- 对于接用户前的运送距离，在查询前直接算好。接用户后的运送距离，可以根据之前算好的距离进行剪枝：如果运送的距离已经使得绕路距离大于10KM时放弃方案。
- 提前计算每个出租车内所有乘客目的地之间的距离。在加入此优化后，计算接用户后的运送距离只需要再提供用户的起点、终点到其它乘客目的地之间的距离。
- 对于每次查询，利用传统的最短路算法（堆优化Dijkstra）计算该用户起点、终点到其它节点的距离。加入此优化后，查询时不再需要进行任何GTree搜索，效率得到了极大提高。
- 尝试了部分启发式算法搜索运送乘客顺序，如每次运送离当前位置距离最短的乘客。

在加入了所有的以上优化之后，经测试，对于一般的例子可以瞬间给出解，而对于测试所用的例子则也能在2秒内求解。由于速度已经相当之快，本人将所有满足的出租车都求解了出来，并对其按照绕路距离进行了排序，也全部输出到了前端。

web设计

web的设计采用了Django框架，而由于其是通过Python写成，和之前的代码不同，故需要考虑Python和C++两个跨语言程序之间的交互问题。本人的做法是由Python建立并运行C++子进程，然后通过文件进行通信。前端收到查询后会向文件内写入信号值，然后由后者读出并运行。运行完成后，C++再向Python返回自己的结果。

对于网页前端部分，使用了JavaScript来显示结果，并通过百度地图的API对所有的出租车位置进行了展示，同时还输出了最优出租车到用户起点的距离。

5. 实验总结

本次实验是数据库的大实验，因此从难度上来说理应比之前的作业更加困难。然而，由于算法最核心的部分已经由GTree完成，导致实际上作业的难度降低了不少，反倒有大量的工作作用在了前端的开发上。事实上，在实验做到后期的时候本人才发现，如果不使用GTree而只使用最短路算法也是能够以很高的效率独立完成这一次作业的，或许这样的解决方式更加合理，不过两种做法都值得我们学习和思考。

这一次数据库的作业给我的感觉是并没有太多和数据库算法相结合的部分，或许加大算法难度会使我获得更多的收获，不过我还是从中受到了许多训练，也感受到了有关出租车调度算法的困难和复杂程度。这些都对我之后的学习和工作能够提到很大的帮助。