

Database Project Report - 互联网拼车

计 55 陈齐斌 2015011403

1 实验说明

- 问题描述在下发的 `project.pdf` 中
- 代码依赖的第三方库、运行方法以及样例结果见 `README.md`
- 实现的功能有：
 - 在 5s 内（不包括预处理）返回 5 辆出租车，并且绕路距离不超过 10km
 - 实现 UI 可以显示乘客位置、目的地位置、返回结果的出租车位置
 - 考虑了最优乘客送达顺序，UI 中显示行进路线以及车上其他乘客的目的地与路线
 - 基于路网来完成（借助了 GPTree 第三方库）

2 设计框架与算法

2.1 总体框架

- 借助了 GTree 仓库中的 GPTree，可以给定路网数据 `road.cnode`，`road.nedge`，经过建树的预处理，之后只要加载保存的 `.data` 文件，就可以在很短的时间内得到两个 node 间的最短路径长度以及路径经过的结点列表。
- 考虑到系统的效率，除了上述的 gptree 用了原来的 c++ 实现外，寻找出租车等函数也用了 c++ 实现。然后用 python 对这些封装后的 api 进行调用，处理用户输入，生成 ui 等操作。

2.2 模块简要文档

2.2.1 core.cpp

- `core.cpp` 中为 GPTree 的算法实现代码，以及搜索出租车相关的代码，细分的介绍如下：
- GTree 两点最短路查询第三方库的部分
 - `load`，`save` 为预处理导出导入的接口
 - `int G_Tree::search(int S, int T)` 为返回两点间最短路径的距离，另外还有 `search_catch` 函数可以在多次连续查询时利用计算的缓存提高效率。

- `int G_Tree::find_path(int S, int T, std::__1::vector<int> &order)` 为返回两点间最短路径经过的 node 的列表（存在 order 中）
- 打车拼车部分
 - `json searchTaxi(int S, int T, int K)` 为返回 K 辆去接从 S 到 T 的乘客的出租车。返回一个 json 变量，其为一个 K 个元素的列表，每个元素为找到的符合条件的一辆出租车的信息字典，key 包含 D1, D2, D3, D4 的值，以及出租车的 id，以便调试与分析。用到了 BFS，即从 S 点出发向四周找车，以便能在最快时间内让用户上车
 - `int optimalOrder(std::__1::vector<int> passengers_pos, int taxi_pos)` 函数，由于不同的接送顺序会产生不同距离的总路程，通过一个全排列来枚举，找到最优的乘客送达顺序下的总路程（也就是 D3），对应还有一个函数 `optimalOrderRoute`，这会返回这个最优路线，是为了尽可能减少不必要的操作而设计。与 `search`，`find_path` 的关系类似。
 - `std::__1::vector<int> wholePath(std::__1::vector<int> nodes)` 由于 `optimalOrderRoute` 返回乘客的送达顺序，并不能提供直接的导航，还需要调用 `find_path` 来找到完整的路径，`wholePath` 即实现这个功能的函数。
 - `int main(int argc, char **argv)` 加载预处理过的内容后，开始循环等待命令行的输入，为 python 提供函数接口

2.2.2 pipe.py

- 实现的 python class `Pipe` 用来开启 c++ 子线程以运行 `core`，提供 `send`，`recv` 两个接口来通过 `stdin` 和 `stdout` 与 `core` 程序通信

2.2.3 api.py

- 通过调用 `pipe`，与 c++ 端对接，将用 `pipe` 实现的 c++ 的函数封装成 python 中的函数，使得 `search.py` 中能正常调用

2.2.4 search.py

- 提供服务的主要代码
- `load_nodes`，`load_taxis` 加载了路网节点信息
- `nearest_node` 通过调用 `distance`，使得能够支持由一个经纬度找到最近的路网节点，以开始之后的主算法。为了尽可能减少重复计算，将一些重复的计算结果缓存下来，使速度提高了很多。
- `main` 中循环等待用户输入经纬度，并将结果写到 `index.html` 网页中，可以打开看到用户位置、出租车位置、最短路线等。

3 参考资料

- `lect06-spatialdata.ppt` 课件

- [TsinghuaDatabaseGroup/GTree](#)
- [Baidu Map 开放 API](#)