Map-walker 软件-课程设计总结

--刘含

学期初，数据结构课程设计布置下来了，题目是“模拟旅行系统”。看着貌似难度也不是太大，只需要实现三种路径策略就行，后面真正开始做的时候便会发现问题重重，诸如换乘时间、发车时间、多终点城市等均不能直接采用 dijkstra 策略，课程设计中便是一个不断发现问题和解决问题的过程。

第一个遇到的是开发语言的选择。因为已经做了一些软件的规划，准备采用服务器和客户端和工作方式，故在服务器端选择了更为方便快捷的 Python3，其中也不乏有对学习一门新语言的好奇。在客户端由于是图形化界面，故采用了更为熟悉的 C++做 QT 开发。

选择好语言后，下一步就是要做架构的设计，这一部分主要是由牛天睿负责，几次会议的讨论，有了较为一致性的看法，分工后便投入了第一版的开发中。

我负责的是数据库、日志和三种路径策略的开发。第一步完成的就是数据库，这块采用的是 sqlite3 轻便型软件数据库。第一版中对数据库预处理后得到的还是分为飞机、火车和汽车的三个二维列表，并且只包含了时间、花费等信息，在后期计算策略时便会显现出很大的不足，信息的不完备、列表的复杂均给路径的计算造成了十分大的困难。第二版便决定修改数据结构，设计了一个路径类，代表一个直接路径，包含了起点、终点、车次、时间、花费等等信息。并且在处理数据库时将飞机、火车和汽车融合到一个二维列表中，其中每一项是两个城市至今所有直接路径的 list。事实证明，这样的修改确实发挥了很大的作用。

做完数据库后并没有直接去开发策略，由于策略那块的多终点没有解决，便转向先做日志了。日志这块相对比较简单些，主要实现了三种日志信息的记录：

warn、debug 和 info，封装好相关函数便可以使用。另外，这块设计了日志自动转储的功能，考虑到服务器端日志文件长期重复使用可能造成过大打开困难等问题，因此当日志文件超过一定行数时便重命名存储到另一个位置。

经过一段时间的考虑，算法策略这块也有些想法了。在前两个策略中，对于多终点城市均采用了相同的策略。即在 dijkstrta 算法中，当加入一个新的结点时，首先判断这个结点在不在终点城市列表中，如果在，那么无条件将所有其他城市的路径修改为经过该终点城市，这样便可以保证找到最后一个终点城市计算的路径包含了用户所要求的所有城市。第一版便这样结束了。第二版中加入了发车时间，多另一个考虑因素，经过上面的数据库模块的修改，则在策略中新加入的一个变量专门用于存储到达城市的时间，多次尝试修改这个问题便也就轻松解决掉了。

第三个策略一直是一个难题，这样的有限制最优看着很像是用动态规划来做，搜索的话时间复杂度感觉有点太高。因此长期以来，不断尝试了 DP 的各种策略，期末验收临近，迫于时间的压力，最终不得已改到了 BFS 上。因为是树形搜索，节点数会成指数形式扩展，便在建树中多次做了限制优化，尽量减少时间复杂度，但还是因为算法本身和 python3 的性能劣势造成了比较大的时延。这块的话后期还有优化空间。

经过这一次的数据结构课程设计，相信自己的编程能力、重构能力、设计能力都得到了相当大的提升。学习了一门新的语言，边学习边开发，慢慢适应着这种过程，我想也许最重要的莫过于此了吧，有了这样的学习能力，自己对于一个新的、陌生的东西敢于去尝试、敢于去探索，并能在痛苦中坚持下来，那么，任何问题都不是问题，都将成为你进步道路上的垫脚石而已。

一个学期的开发之路，回忆起来，充实、满足。

作者：刘含

2016/6/4