

## PJ3 最短路径算法实现上海地铁最佳线路选择 项目开发文档

20307140059 艾匡时

### Part 1 算法基本思路

由于需要实现的目标是输出给定起点和终点之间耗时最短的路径，选用了单源最短路径算法，用 Dijkstra 算法进行实现。同时，本项目中的图采用邻接表来表示。与常规的 Graph 类及 Dijkstra 算法不同的是，本项目因为需要在换乘站提示用户，而其他路径可以隐藏，故 Graph 类的边界点实现时特意加入了表示边在哪一条地铁线的私有变量，方便输出时使用。此外，地铁站名是中文字符串，而 Graph 类的顶点名用整型数字会更加方便，因此引入了 `vector<string>` 容器，用于存储地铁站名，并将它们的下标与图中的站点名一一对应。

总体而言，本次项目的算法思路是基础且明确的，但是考虑到具体应用场景，功能细节上的增补较为繁琐。

### Part 2 开发中遇到的问题及解决

最大的困难就是如何把 Excel 表格里面的数据输入程序，起初打算采用读取文件的方式，但是 C++ 对于 Excel 表格的读取非常繁琐，方便找到的库只能读取 csv 格式的文件，没有 Python 一般的简便。尝试过后，我最终使用 Excel 先把数据处理完毕（也因而学会了很多 Excel 的内置函数用法），再复制导入程序内用 `vector<string>` 读取。Excel 的功能比我想象中要强大很多，处理看似繁琐的地铁发车表绰绰有余。当然这个过程依旧耗费了不少精力。

此外，由于我之前没有修过 C++ 面对对象的编程，我对 C++ 中类

的使用较为生疏，本项目也花了不少时间完善自定义的 Graph 类。不过有了前面 project 的练习，这次已经熟练了很多。

### Part 3 可能的优化方向与功能的完善

在 Dijkstra 算法中，选取 Dist 表中最小元素，如果采用最小堆来实现，可以将时间复杂度降低一个数量级，可以作为一个主要的优化方向。

本项目没有考虑换乘所需时间和各条线路的始末发车时间，因此很多情况下的输出结果是没有现实指导意义的。如果考虑相对时间和绝对时间，采取时间复杂度更优的单源最短路径算法，可以作为导航算法。