广州地铁线路查询程序的实现

王若琪　　18340166　　wangrq29 @mail2.sysu.edu.cn　　2019/12/24

本实验由我与吴晓淳同学（wuxch8 @mail2.sysu.edu.cn）合作完成。

一、实验目的

设计并实现一个地铁线路查询程序，输出最佳乘车方案等信息，依次来熟悉无向图有权无权最短路径的算法并锻炼实际应用能力。

二、实验环境

本实验在 Windows 10 下完成。

- 开发工具：Dev-C++
- 编程语言：C++

三、分析与设计

3.1 需求分析

- 要求实现一个地铁线路查询程序，题目要求最少一种，而我们能给出3种最佳乘车方案，分别是最短里程、最少换乘、最小站数。
- 我们设计的程序能够查询广州地铁的所有路线和车站（路线数据来源于2019年12月23日时广州地铁官网 （www. gzmtr.com) 更新数据，但不包括广佛线和APM线）。
- 要求设计尽量友好的用户界面。

3.2 数据结构设计

- 路线结构：

```
struct{
    int Lnum;//线路编号
    int stationNum; //站台数
    vector<string> v;//站名
    vector<int> w;//weight
}Line[22];
```

- 顶点结构：

```
struct vertex{
    bool known; //标记是否已知
    int dist; //记录已知最小距离
    int path; //记录上一个顶点编号
};
```

- 带权图结构，用来算最短里程的乘车方案：

```
struct Graph_a{
    int vexnum; //顶点数
    int arcnum; //边数
    vector<vertex> v; //所有顶点向量
    vector<vector<pair<int,int> > > adj;//邻接表 int 编号 int 权重
};
```

- 无权图结构，用来算最小换乘和最小站数的乘车方案：

```
struct Graph_b{
    vector<vector<int> > adj;//邻接表存图
    int vexnum;//顶点数
};
```

3.3 算法设计

1. 基于Dijkstra算法的求最短里程乘车方案的方法：

   (1) 由于广州地铁App没有给出相邻地铁站间的具体距离，但有提供往来相邻地铁站所需的时间，因此在此假设各条线路地铁的平均速度相同，根据公式 s = v * t 可以以时间代替距离来进行计算；

   (2) Dijkstra算法:

   为节省空间，假设用带权的邻接表adj表示带权无向图，adj[i].at(j).second 表示弧<Vi, Vadj[i].at(j).first >上的权值。若<Vi, Vadj[i].at(j).first >不存在，则置adj.v.dist为INT_MAX，S为已找到从v出发的最短路径的终点的集合，它的初始状态为空集。那么从v出发到图中其余各顶点（终点）vi可能达到的最短路径的长度的初值为：v[adj[i] [j].first].dist = adj[i] [j].second　vi ∈ V

   选择vj，使得 v[j].dist = Min{ v[i].dist | vi ∈ V-S }vj 就是当前求得的一条从 v 出发的最短路径的终点。令S = S ∪ { j }即置 v[j].known = true。修改从 v 出发到集合 V-S 上任意顶点 vk 可达的最短路径长度。如果adj[j] [k].second + v[j].dist < v[adj[j] [k].first].dist ，

   则修改v[adj[j] [k].first].dist 为v[adj[j] [k].first].dist = adj[j] [k].second + v[j].dist。重复操作，由此求得从 v 到图上其余各顶点的最短路径是依路径长度递增的序列。

2. 基于BFS求无权图最短路径算法的，求最少换乘数乘车方案以及最少站数乘车方案的方法：

   （1）最少站数：

   求最少站数的乘车方案相当于求无权图的最短路径，我们采用的是BFS的方法，先寻找据起始站距离为1的站，判断这些站里有没有目的地，如果有，则找到最短路径，如没有，则继续寻找据起始点距离为2的站，继续判断这些站里边是否有目的地……每次记录这个点的上一个点，以便输出路径。以此类推，最终可以求出站数最少的路径。

   （2）最少换乘数：

   为了使方法尽量简单，综合地铁线路的特点，只需要在制图过程与最少站数方法稍作区别，其余过程都与最少站数的BFS方法一致。在制图过程中，因为一条线路上的所有站点之间都不需要换乘，所以可以将在一条线路上的每一个站点都看做是一个点来制图。具体方法就是，在一条线路的任意两点之间都连上边，这样生成的图用BFS方法算出的最短路径长度即为最少换乘数加1，求出路径上的站点即为需要换乘的站点。

3.4 代码主模块命名清单

源代码中除 `main()` 函数之外的各个函数命名清单如下：

- `read`函数功能有，读取文件内容，并将路线编号、站点名称、站间距离等信息提取并整合到各种数据结构中：

```
void read(int i,string fileName,int &cnt);
```

- `mkGraph_a`函数的功能是形成`Graph_a`类型的图：

```
Graph_a mkGraph_a();
```

- `mkGraph_b_c`函数的功能是形成`Graph_b`类型的图，此函数能够生成适用于找到最小换乘方案的算法，即在一条线上的任意两点之间都有一条边：

```
Graph_b mkGraph_b_c();
```

- `mkGraph_b_s`函数的功能是形成`Graph_b`类型的图，此函数能够生成适用于找到最少站台数乘车方案的算法，与`mkGraph_b_c`不同的是，此函数生成的图是正常的地铁线路图：

```
Graph_b mkGraph_b_s();
```

- `judge`函数可以读入起点和终点，并判断站名是否合法：

```
bool judge(string &name1, string &name2)
```

- `MinDist`实现用Dijkstra算法找最小里程的路径：

```
void MinDist(string v1,string v2,Graph_a g);
```

- `find_min`实现用BFS方法，根据传入的图，找最小换乘或最小站数的路径：

```
void find_min(Graph_b g, int start, int dest, bool ifChange)
```

- `void MinDist_`函数的功能是读入起点和终点，调用`judge`函数和`MinDist`函数：

```
void MinDist_(Graph_a g);
```

- `void MinChange_`函数的功能是读入起点和终点，调用`judge`函数和`find_min`函数，并且向find_min传参时ifchange为true：

```
void MinChange_(Graph_b g);
```

- `void MinStation_`函数的功能是读入起点和终点，调用`judge`函数和`find_min`函数，并且向find_min传参时ifchange为false：

```
void MinStation_(Graph_b g);
```

- `print_whole`函数负责显示所有路线和站点：

```
void print_whole();
```

- 以下3个函数实现3种结果路径的输出：

```
void print_a(int i,Graph_a g);
void print_min_s(vector<int>from, int dest, int start);
void print_min_c(vector<int>from, int dest, int start);
```

四、程序测试与使用结果分析

我随机做了对不同线路上站点的测试，并经过人工检查，发现符合实际要求。下面以从大学城北到天河客运站以及广州南站到机场北（2号航站楼）为例来展示测试结果。

【注：路线数据来源于2019年12月23日时广州地铁官网 （www. gzmtr.com) 更新数据，但不包括广佛线和APM线，并且这里0号线表示3号线北延段，15号线表示14号线支线（知识城线）。】

图(1)为从大学城北到天河客运站的最短里程数的乘车方案输出结果：

```
请输入出发地：大学城北
请输入目的地：天河客运站

最短距离的乘车方案有：
大学城北->官洲->万胜围->车陂南(换乘 5 号线 往 滘口 方向列车)->科韵路->员村->潭村->猎德->珠江新城(换乘 3 号线 往 天河客运站 方向列车)->体育西路(换乘 0 号线 往 机场北（2号航站楼） 方向列车)->林和西->广州东站->燕塘(换乘 6 号线 往 香雪 方向列车)->天河客运站
```

图(1)

图(2)为从广州南站到机场北（2号航站楼）的最短里程数的乘车方案输出结果：

```
请输入出发地：广州南站
请输入目的地：机场北（2号航站楼）

最短距离的乘车方案有：
广州南站->石壁->会江->南浦->洛溪->南洲->东晓南->江泰路->昌岗->江南西->市二宫->海珠广场->公园前->纪念堂->越秀公园->广州火车站->三元里->飞翔公园->白云公园->白云文化广场->萧岗->江夏->黄边->嘉禾望岗(换乘 0 号线 往 机场北（2号航站楼） 方向列车)->龙归->人和->高增->机场南（1号航站楼）->机场北（2号航站楼）
```

图(2)

图(3)为从大学城北到天河客运站的最少转乘数的乘车方案输出结果：

```
请输入出发地：大学城北
请输入目的地：天河客运站

最少转乘数的乘车方案有：
大学城北-[沿4号线一直到]->车陂南-[沿5号线一直到]->珠江新城-[沿3号线一直到]->天河客运站
共换乘2次
```

图(3)

图(4)为从广州南站到机场北（2号航站楼）的最少转乘数的乘车方案输出结果：

```
请输入出发地：广州南站
请输入目的地：机场北（2号航站楼）

最少转乘数的乘车方案有：
广州南站-[沿2号线一直到]->嘉禾望岗-[沿0号线一直到]->机场北（2号航站楼）
共换乘1次
```

图(4)

图(5)为从大学城北到天河客运站的最少站台数的乘车方案输出结果：

```
请输入出发地：大学城北
请输入目的地：天河客运站

最少站台数的乘车方案有：
大学城北->官洲->万胜围->车陂南->科韵路->员村->潭村->猎德->珠江新城->体育西路->林和西->广州东站->
燕塘->天河客运站
```

图(5)

图(6)为从广州南站到机场北（2号航站楼）的最少站台数的乘车方案输出结果：

```
请输入出发地：广州南站
请输入目的地：机场北（2号航站楼）

最少站台数的乘车方案有：
广州南站->石壁->谢村->钟村->汉溪长隆->大石->厦滘->沥滘->大塘->客村->广州塔->珠江新城->体育西路->
林和西->广州东站->燕塘->梅花园->京溪南方医院->同和->永泰->白云大道北->嘉禾望岗->龙归->人和->高增
->机场南（1号航站楼）->机场北（2号航站楼）
```

图(6)

五、小组分工

1. 先经过讨论，确定出整体设计思路。共同写出所有需要用到的数据结构，并讨论确定数据统计方法与途径。
2. 吴晓淳同学负责数据整理，并完成程序代码中读取文件并整合的部分。
3. 吴晓淳同学负责完成有关最小里程数乘车方案的实现，以及所有相关的函数如生成图和输出路径的实现。
4. 王若琪同学负责完成有关最少换乘数乘车方案、最少站台数乘车方案的实现，以及所有相关函数如生成图和输出路径的实现。
5. 王若琪同学负责函数接口整理以及用户界面的实现。

六、不足与问题

由于能力不足和时间问题，我们发现我们的程序存在一处不完美的地方，即如果以某一种方式去求解最优路径存在2种及以上个结果时，我们只能输出最先得到的一种，而不是输出所有最好结果，或是在这些最好结果中，再用其他标准进行比较再筛选。由于期末考在即，我们无暇再去完善，但我们将会在期末后继续完善程序，改进这一不完美之处。

七、反思与总结

1. 这次大作业的设计比前几次过程顺利很多，我们两人大约用了两天时间就基本搞定了。这既是因为作业内容比前两次的简单许多，也是因为我们已经经历过很多次磨合，对合作流程都有了一定的经验，所以很快的分工好，顺利完成作业。
2. 在设计时遇到了中文文件读取乱码的问题，经过查阅资料，发现是由于汉字编码方式不同产生的问题，于是将存放地铁线路的文本文件编码方式都改为ANSI，就能够正确读取了，这是我从这个实

验中学到的新方法。
3. 总而言之，这次作业的完成很顺利，我将从这次实验中总结合作经验，争取在以后的程序设计中都保持这样的状态。

八、参考资料和网页

[1]Mark Allen Weiss 数据结构与算法分析——C++语言描述（第四版）中文版 P308~P316

[2]广州地铁官网 （www. gzmtr.com）