Shortest Path Problem

詹江岳

Date: 2017-12-21

Chapter 1: Introduction

问题描述：用一个有向图表示给定的 n 个（要求至少 10 个）城市（或校园中的一些地点）及其之间的道路、距离情况，道路是有方向的。要求完成功能：根据用户输入的任意两个城市，给出这两个城市之间的最短距离及其路径。

算法背景：寻找两地间的开销最小的路径，具有重要的现实意义。旅游出行，上下班等，如果能走开销最小的路径，那就能省下不少钱。

Chapter 2: Algorithm Specification

- 主要数据结构设计说明

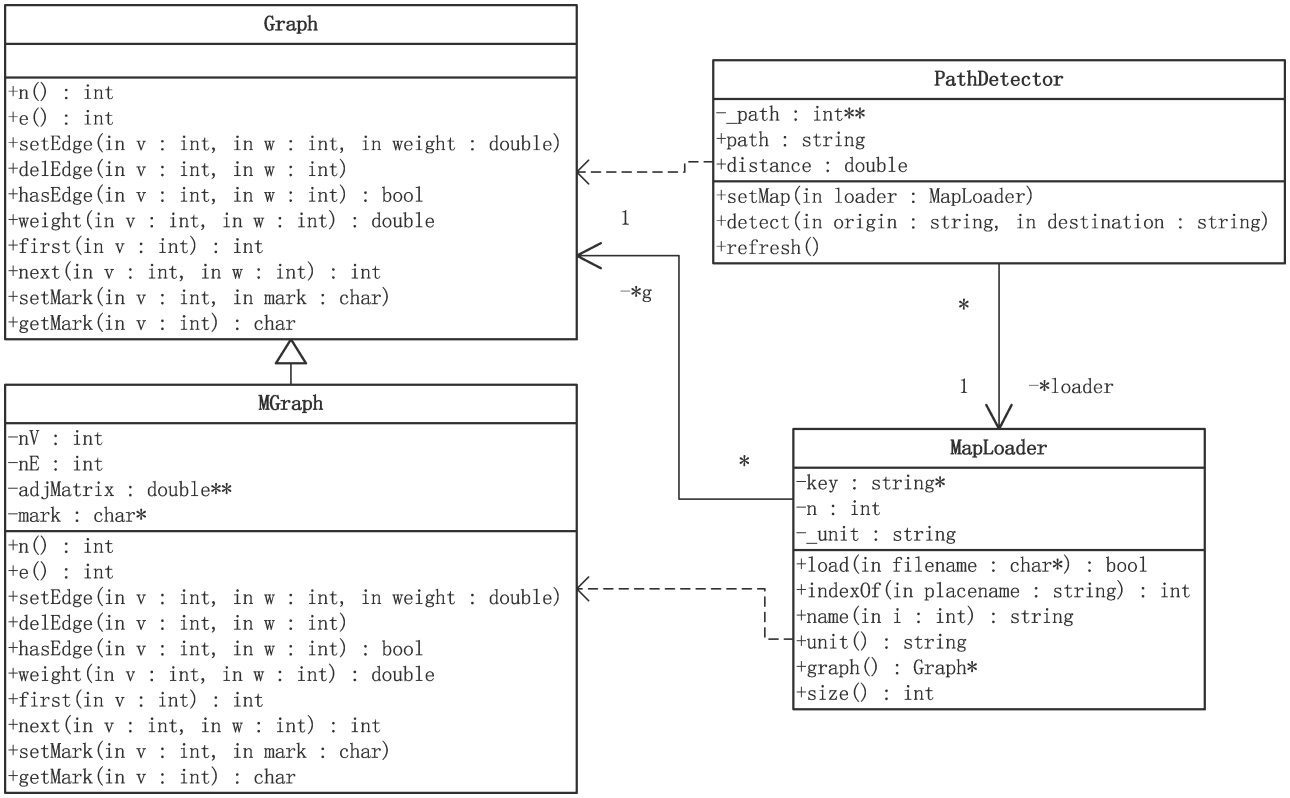
  Graph

  逻辑上由顶点和连接顶点的边组成，采用矩阵（二维数组）实现。

  每两个顶点间可以存在边，也可以不存在边。边是有向的，带有权值。如果存在从顶点 i 到顶点 j 的权值为 w 的边，则存储矩阵第 i 行、第 j 列元素的取值为 w；否则为 0。

主要操作	方法说明
int n() const	获取顶点数目
int e() const	获取边的数目
void setEdge(int v, int w, double weight)	设置从 v 到 w 的权值为 weight 的边
void delEdge(int v, int w)	删除从 v 到 w 的边
bool hasEdge(int v, int w) const	判断是否存在从 v 到 w 的边，存在则返回 true
double weight(int v, int w) const	获取从 v 到 w 的边的权值
int first(int v) const	获取与 v 邻接的点的集合中的第一个元素
int next(int v, int w) const	获取与 v 邻接的点的集合中元素 w 的下一个元素
void setMark(int v, char mark)	设置顶点 v 的标记为 mark
char getMark(int v) const	获取顶点 v 的标记

- 系统设计思想：



- 程序流程图

  见附件《附件 1_流程图.png》

Chapter 3: Testing Results

模块	用例	用例描述	预期结果	实际结果	错误估计	现状
强连通图 cg.png	载入cg.txt	加载地图信息	载入文件，进入查询页	载入文件，进入查询页		pass
	查询A到C	查询可达的两点	ABC 12	none INFINITY	计算最短路径时出错，distance[i][j]和distance[i][k]+distance[k][j]比较时未考虑distance[i][j]为INFINITY的特殊情况	pass
	查询C到A	查询可达的两点	CA 1	CA 1		pass
	查询A到FF	查询可达的两点	ADE,FF 16.6	ADE,FF 16.6		pass
非连通图 ucg.png	载入ucg.txt	加载地图信息	载入文件，进入查询页	载入文件，进入查询页		pass
	查询C到A	查询可达的两点	CA 1	CA 1		pass
	查询A到FF	查询不可达的两点	none INFINITY	none INFINITY		pass
平凡图 tg.png	载入tg.txt	加载地图信息	载入文件，进入查询页	载入文件，进入查询页		pass
	查询A到A	查询到自身的路径	A 0	A 0		pass
	查询A到FF	查询到不存在的点的路径	none INFINITY	none INFINITY		pass
完全图 cpg.png	载入cpg.txt	加载地图信息	载入文件，进入查询页	载入文件，进入查询页		pass
	查询A到D	查询可达的两点	AD 5.6	AD 5.6		pass
	查询D到C	查询连接两点的边不是最短路径时的最短路径	DAC 7	DAC 7		pass
零图 ng.png	载入ng.txt	加载地图信息	载入文件，进入查询页	载入文件，进入查询页		pass
	查询A到D	查询不可达的两点	none INFINITY	none INFINITY		pass
	查询D到C	查询不可达的两点	none INFINITY	none INFINITY		pass
空白(没有图)	载入nerr.txt	试图载入顶点数标记为0的文件	报错	报错		pass
实际场景 rg.png	载入rg.txt	加载地图信息	载入文件，进入查询页	载入文件，进入查询页		pass
	查询ERji到Ylcan	查询可达的两点	ERji-FAxueyuan-Ylyun-Ylshiyanlou-Ylshuxueyuan-Yican 4080m	ERji-FAxueyuan-Ylyun-Ylshiyanlou-Ylshuxueyuan-Yican 4080m		pass
	查询JIANhuan到TUshuguan	查询可达的两点	JIANhuan-ERji-TUshuguan 1190m	JIANhuan-ERji-TUshuguan 1190m		pass
文件	载入err.txt	试图载入空文件	报错	报错		pass
	载入不存在的文件null.txt	试图载入不存在的文件	报错	报错		pass

Chapter 4: Analysis and Comments

- 算法分析：

  时间复杂度：对于有 n 个顶点的有向图，初始化矩阵是 n^2 的操作，计算最短路径的过程需要 n^3 次操作，因此整个生成最短路径矩阵的过程的时间复杂度为 $\Theta(n^3)$。查询两点间的最短路径只要查询矩阵的某一行，因此时间复杂度为 O(n)。

  空间复杂度：图用邻接矩阵存储，因此复杂度为 $\Theta(n^2)$。最短路径矩阵的存储也是 $\Theta(n^2)$。此外，生成最短路径矩阵时所用的最短距离矩阵的复杂度也是 $\Theta(n^2)$。因此算法整体的空间复杂度为 $\Theta(n^2)$。

- 算法特色：

  对一个顶点数为 n 的图 g，连续的 m 次查询的时间复杂度是 $O(n^3+mn)$，查询次数越多（m>n），平均每次查询的时间开销越少。

- 不足：

  如果图 g 的邻接矩阵是稀疏矩阵，那么用二维数组的实现方式空间浪费较大。如果对一个图，只进行几次查询，那么 $O(n^3+mn)$的时间复杂度无法显现平均上的优势，此时直接用 Dijkstra 算法更好（复杂度为 $\Theta(mn^2)$；用堆优化后是 $\Theta(m(e+n)\log n)$，e 为边数；堆的实现可以在实验 *Huffman Compression* 里找到）。可以考虑增加对图顶点数和边数的判断，动态决定使用邻接矩阵还是邻接表存储图。可以考虑增设查询模式，针对用户预期的不同查询情况，采用不同的算法。

  此外，载入文件的格式上，可以考虑增加[起点 终点 距离]这样的格式的输入，减少一些写邻接矩阵时带来的麻烦。可以考虑增加文件格式检查（含地点重名检查等），提高程序的安全性。

  目前尚不支持中文输入，可以考虑采用更好的技术，或改变交互方式来解决。

Declaration

I hereby declare that all the work done in this project titled "Shortest Path Problem" is of my independent effort as an individual.

Duty Assignments:

Programmer: 詹江岳

Tester: 詹江岳

Report Writer: 詹江岳