|  |
| --- |
| 哈尔滨工业大学(深圳) |
| **《数据结构》实验报告** |
|  |
| 实验四  图形结构及其应用  学 院:   |  |  | | --- | --- | | 姓 名: |  | | 学 号: |  | | 专 业: |  | | 日 期: |  | |

# 问题分析

1.任务一：

（1）实验要求建图，这个已经有代码了

（2）实验要求判断图是否连通，只需进行DFS遍历，若整个图都能遍历到则说明图为连通。

（3）实验要求计算每个节点的度，只需对其临界矩阵的每一行进行遍历，记录连通点即可。

（4）实验要求计算图的聚类系数。点的聚类系数是所有与它相连的顶点之间所连的边的数量，除以这些顶点之间可以连出的最大边数。而图的聚类系数是所有点的聚类系数的均值。需对每个点记录其邻居，并求出其邻居间可能相连的条数，即求出（n为邻居数），再求出邻居间实际的连线条数，即可算出一个点的聚类系数。求均值即可求得图的聚类系数。

（5）实验要求运用Dijkstra算法计算最短路径，直接运用Dijkstra算法即可。

（6）实验要求计算图的直径，半径。直径，半径分别为图的最大和最小的偏心率（Eccentricity），而偏心率（Eccentricity）为从任意一个节点，到达其他节点的最大节点距离。节点距离即为两个节点间的最短路径的长度。需运用Floyd算法求出图中每个节点到其他节点的最短路径，然后求出每个点到其余点最短路径中的最大值，即为这个点的偏心率（Eccentricity）。求出图的偏心率（Eccentricity）的最大最小值，即可求得图的直径与半径。

2.任务二：实验二分别要求对深圳地铁线路图进行一下操作

（1）求线路图是否连通。

（2）求线路图中换乘线路最多的站点与通过其的线路。

（3）求线路图的直径和半径。

（4）求从大学城站到机场站最少需要的时间，并打印最短路径。

对与（1），（2），（3），（4），可对应到任务一的（2），（3），（5），（6），即用任务一的代码即可完成任务。

（注：任务一与任务二分开在两个文件中）

# 二、详细设计

## 2.1 设计思想

任务一：

1. 判断图是否连通，这里采用DFS（深度优先遍历）对图进行遍历，并且对遍历到的点进行标记。遍历完成后，若存在某点未被标记，即未遍历到整个图，即图不连通。若所有点都被标记，则表明图连通。
2. 对系数矩阵每一行进行遍历，记录此行中连通项的个数，即为此节点的度，对所有行遍历后，即可求得图中每一点的度。
3. 首先用一个二维数组记录每个节点的所有邻居，然后求出每个节点的邻居数，用一维数组储存。再判断每个节点的邻居之间两两是否有连接，记录所有的连接数，储存在一维数组中。接下来进行聚类系数的计算。若一个节点的邻居小于等于1，则其聚类系数自然为0，无需进行计算。若一个节点的邻居数大于等于2，则计算 即为此点的聚类系数，储存到一维数组中。计算所有点的聚类系数平均值即为图的聚类系数。
4. 运用Dijkstra算法求单源最短路径。进行如下循环：目标起点A和选定点B之间，若存在另一点C，使得，则将C标记，修改A，B间的最短路径长度，对除了A，B外所有点进行此遍历。将选定点B遍历过除了A外所有点后，即完成一次循环。此次循环求得A与其余点中一点的最短路径。重复此循环N-1（N为节点数）次后即可求得A到其余点的最短路径。进行循环时，若B为所求目标终点，则将每个选出的C加入到路径path中，并且令path的第一位元素为start，最后一位元素为end，即可打印初最短路径。
5. 首先运用Floyd算法求出所有点之间的最短路径的长度：取一点C，若有A，B，，则修改A，B间的最短路径。B，C将所有点遍历后，中途经过A点的最短距离以算出，再将A进行遍历，即可算出所有点之间的最短路径的长度。对每个点到其余点之间的距离求最大值，即为偏心率（Eccentricity）。对所有点的偏心率求最大值，最小值，即得图的直径和半径。

任务二：

任务二可直接运用任务一中的函数完成，算法思路与任务一相同，无需赘述。

（注：任务一与任务二分开在两个文件中）

## 2.2 存储结构及操作

(1) 存储结构（一般为自定义的数据类型，比如单链表，栈等。）

图节点Graph，包含顶点数N（整型），边数E（整型），邻接矩阵matrix（整型二维数组），节点名字vertex（字符串）。

(2) 涉及的操作（一般为自定义函数，可不写过程，但要注明该函数的含义。）

|  |  |  |  |
| --- | --- | --- | --- |
| 函数名 | 参数 | 功能 | 返回值 |
| createGraph | 节点数n | 建图 | 图节点g |
| printPath | 路径长度d，路径指针path，图节点g | 打印最短路径 |  |
| isConnected | 图节点g | 判断图是否连通 | 布尔运算值0/1 |
| DFS | 图节点g，起点标志i | 从i开始深度优先遍历图g |  |
| nodeDegree | 图节点g，度数组node\_degree | 求图的每个节点的度 |  |
| clusteringCoefficient | 图节点g | 求图的聚类系数 | 聚类系数aver\_clco |
| dijkstra | 图节点g，起点位置start，终点位置end，路径指针path | 用dijkstra算法计算单源最短路径 | 最短路径长度 |
| computeEcc | 图节点g，直径指针diameter，半径指针radius | 计算图的直径和半径 |  |

## 2.3 程序整体流程（略）

# 三、用户手册

任务一：

程序接受数据的方式为从一个名为“stu.in”的程序中读取

输入格式如下：

（1）首先输入图的节点数，然后换行

（2）再输入图的边数，然后换行

（3）依次输入：节点1（空格）节点2（空格）边的权值，然后换行

（4）若边数未到输入的要求，重复（3）

（5）若边数已达到输入要求，但需要再输入其他图，重复（1）~（4）

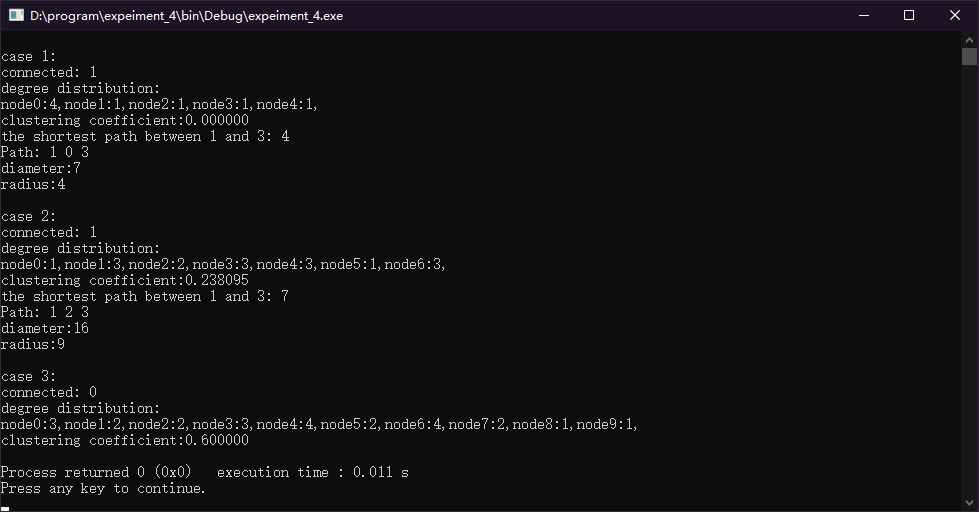
（6）若不输入其他图，直接结束输入即可。

任务二：

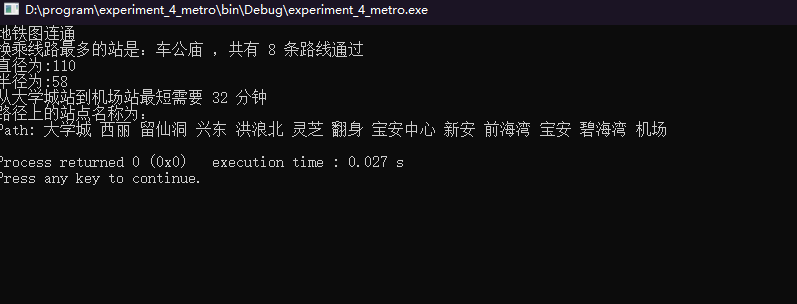
1. 程序从一个名为“no2metro.txt”的程序中读取站点编号和名字
2. 首先输入节点数量，然后换行
3. 依次输入：节点编号（空格）节点名称，然后换行
4. 若未达到输入的节点数量，重复（2）
5. 若已达到输入的节点数量，结束输入
6. 程序从一个名为“metro.txt”的程序中读取站点线路信息
7. 首先输入节点数量，然后换行
8. 输入地铁线路，然后换行
9. 输入此线路数量数量，然后换行
10. 依次输入：站点编号（空格）从上一站到这一站的时间，然后换行
11. 若站点数量未达到输入的此条线路的站点数量，则重复（4）
12. 若地铁线路数量未达到输入的地铁线路数量，重复（3）（4）（5）
13. 若地铁线路数量已达到输入的地铁线路数量，结束输入

# 四、结果

任务一：



任务二：



# 五、总结

该实验涉及到的数据结构和算法，以及遇到的问题和收获。

1. 数据结构：图
2. 算法：
   1. 用DFS进行图的遍历
   2. 用二元数组储存节点的邻居，然后以此为依据求出邻居数及邻居间的连线数，从而求出聚类系数
   3. Dijkstra算法求单源最短路径
   4. Floyd算法求所有点之间的最短路径
3. 问题：
   1. 开始时判断图是否连通时直接对邻接矩阵每一行判断是否有连接，未采用DFS判断。
   2. 计算聚类系数时未考虑邻居数为0或1的点的聚类系数即为0，带入公式计算导致除数为0，造成错误。
   3. 开始时对Dijkstra算法不理解，造成写程序时无从下手。
   4. 进行任务二的编写时，由于对文件读取的不熟练造成障碍。
   5. 进行任务二的编写时，储存读取数据的数组长度设置错误导致缓冲区溢出。
   6. 在实验室写完程序未拷贝到自己的电脑上，导致程序丢失，只能重新编写
   7. 变量名字设计不恰当，导致编写程序时忘记变量的作用，造成编写错误
4. 收获
   1. 熟悉了DFS算法，Dijkstra算法，Floyd算法等与图有关的算法
   2. 熟悉了文件的读取操作
   3. 明白了变量名称的设计要清楚明白且合理，不能随意设计
   4. 明白了程序写完后要及时保存与上传。