|  |
| --- |
| 哈尔滨工业大学(深圳) |
| **《数据结构》实验报告** |
|  |
| 实验四  图型结构及其应用  学 院: 计算机科学与技术   |  |  | | --- | --- | | 姓 名: | 邓皓元 | | 学 号: | 200110618 | | 专 业: | 计算机科学与技术 | | 日 期: | 2021-5-20 | |

# 一、问题分析

第一部分

1.判断图是否连通,从一个顶点开始遍历，如果能遍历到所有点，那么图连通。

**构造邻接矩阵并进行节点个数次相乘，若最后得到的矩阵全为1，则图连通；否则，图不连通。**

2.计算图中每个点的度,节点度是指和该节点相关联的边的条数。

**根据与每个点相连的边数来求取每个点的度**

3.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。图的聚类系数是所有点的聚类系数的均值。

**先计算出每个点的聚类系数：计算与该点连接点的数量，并求取这些点之间的边的数量，再利用数字关系求聚类系数**

4.若图连通，使用Dijkstra算法计算单源最短路径。

**运用Dijkstra算法，从开始点出发，每次寻找与当前点最短距离且为被确定的点，并将该点作为下一个当前点确定，若新路径长度小于原路径则进行更新，记录路径则可以利用每个点的前驱点来求取。**

5.若图连通，计算图的直径，半径。节点距离：指的是两个节点间的最短路径的长度，直径：图中的最大的最短路径的长度，半径：图中的最小的最短路径的长度。

**运用floyd算法，构造带权邻接矩阵，用三个for将该邻接矩阵更新节点个数次，每次更新若新路径长度小于原路径则更新，最后得到的矩阵则是关于每个点之间最短路径的矩阵。**

第二部分

提供深圳地铁的线路图，请自行读取文件建图，并回答以下几个问题：

1. 这个图是连通的吗？

**（等价于第一部分1）**

1. 线路图中换乘线路最多的站点是哪个？共有几条线路通过？

**（等价于第一部分2，只是需要在得到的度中找到并记录度最高的节点）**

3. 从大学城站到机场站最少需要多少时间？请打印最短路径上的站点名称。**（等价于第一部分4，只需读取大学城站的序号和机场站的序号，再利用第一部分的Dijkstra算法函数即可）**

4. 该线路图的直径和半径是多少？

**（等价于第一部分5）**

# 二、详细设计

## 2.1 设计思想

第一部分

**1.构建该图的邻接矩阵，设节点数为n。若邻接矩阵进行n次相乘后全为一，则该图是连接图；否则该图不是连接图。**

**2.直接计算与每个点连接的边的数量，将其除以2则是度。**

**3.先计算出每个点的聚类系数：计算与该点连接点的数量，再用一个数组存取这些点，并求取这些点之间的边的数量，再利用数字关系求聚类系数。最后求取点的聚类系数的平均数就是图的聚类系数。**

**4.运用Dijkstra算法，从开始点出发，每次寻找与当前点最短距离且为被确定的点，并将该点作为下一个当前点确定，若新路径长度小于原路径则进行更新，记录路径则可以构造一个新数组，记录每个点的前驱点数组，最后得出路径。**

**5.运用floyd算法，构造带权邻接矩阵，用三个for将该邻接矩阵更新节点个数次，每次更新中若新路径每次更新若新路径长度小于原路径则进行更新，最后得到的矩阵则是关于每个点之间最短路径的矩阵。**

第二部分

**构建地铁线路图主要依靠的是站点和站点之间距离的读取，所以只需要在main函数中保证数据的正确读取，并修改pathprint函数即可。（构建一个新的字符串组，二维字符数组）**

## 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

1. 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。）

Graph createGraph(int n)：创建一个节点数为n的图

void printPath(int d, int \*path, Graph g)：根据距离d和路径数组path输出路径，这样就不需要路径的节点数也能正确输出路径

int isConnected(Graph g)：判断图是否连通

void nodeDegree(Graph g, int \*node\_degree)：计算每个点的度

double clusteringCoefficient(Graph g)：计算图的聚类系数

int dijkstra(Graph g, int start, int end, int \*path)：使用dijkstra算法计算单源最短路径

void computeEcc(Graph g, int \*diameter, int \*radius)：计算图的直径和半径，提示: Floyd算法

## 2.3 程序整体流程（略）

第一部分

1. 在stu.in中输入图的数据
2. 利用数据构建图
3. 判断该图是否连通并显示
4. 计算每个点的度并输出每个点的度
5. 计算出图的聚类系数并输出
6. 运用dijkstra算法计算两个节点之间的最短路径并输出最短路径和最短路径长度
7. 运用floyd算法计算图的直径和半径

第二部分

1. 在metro.txt和no2metro.txt中分别输入线路图的数据（路线长度和站点名字）
2. 判断该图是否连通并显示
3. 利用数据构建线路图
4. 计算出换乘线路最多的站点并输出该站点和该站点的度
5. 计算从大学城站到机场站最少需要的时间，并打印最短路径上的站点名称
6. 计算该线路图的直径和半径并输出

# 三、用户手册

(1)输入数据的方式

**第一部分**

在stu.in中，先输入顶点数，换行，再输入边数，换行，其中一个边的数据输入方式（再输入“顶点”“空格”“顶点”“空格”“边的权值”换行）。

再次输入上述操作来输入新图的数据

**第二部分**

在metro.txt中，先输入站点数，换行，线路数，换行，再依次输入每条线路的边数据（输入线路的边数，换行，之后输入每条边的数据“站点序号”“空格”“站点距离上个站点的距离”，换行）

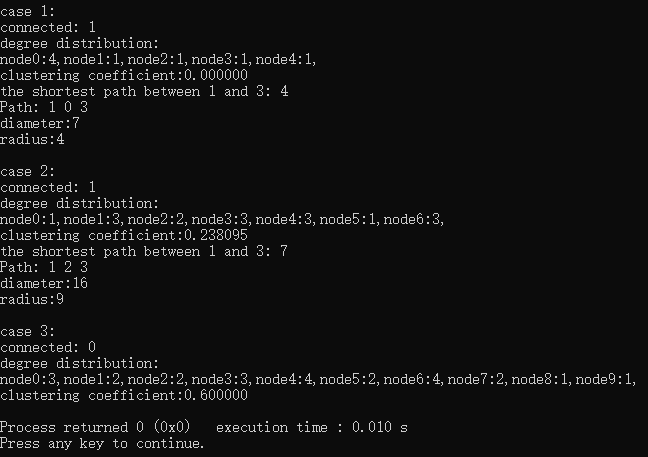
在no2metro.txt中，先输入站点数，换行，再依次输入每个站点的数据（“站点序号”“空格”“站点名字”，换行）

(2)实现各种功能的操作方式等

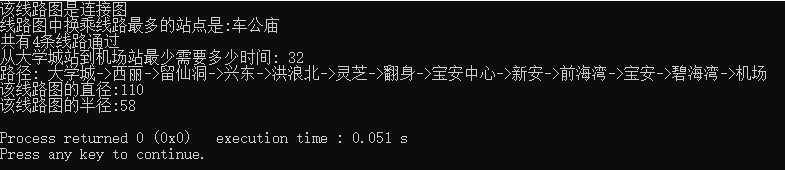
无需输入数据以外的操作即可得知该图的各项数据。

# 结果

**第一部分**

****

**第二部分**

****

# 五、总结

通过这次实验，我对于图这个存储结构有了更加深入的理解，对于有关图的算法和相关知识有了更加透彻的学习和认识，特别是对于dijkstra算法和floyd算法的原理有了更加深刻的认识。除此之外，我还在这次的实验中对于手动加入输出进行调试、字符数组数组、以及指针的用法更加熟练了。

通过这次课程设计，我们对C语言以及数据结构有了更深刻的了解，增强了程序的编写能力，巩固了专业知识。尽管在程序的运行与调试过程中出现了很多错误，通过反复地复习课本上的相关知识，不停地修改与调试，终于完成了这段程序。在调试过程中，我们认识到了数据结构的灵活性与严谨性，同一个功能可以由不同的语句来实现，但编写程序时要特别注意细节方面的问题，因为一个小小的疏忽就能导致整个程序不能运行。我们也认识到了自己的薄弱之处，如对指针相关知识的欠缺，文件读取运用的不熟练，在以后的学习中我们要集中精力、端正态度，争取把知识学得更扎实、更全面。