《数据结构》实验报告

姓名：吴泽同　　学号：084622109
班级：计算机 221　　日期：2023/10/20
程序名：数据结构实验 7：图的构建与遍历

一、上机实验的问题和要求：

1、 **图的邻接矩阵存储与遍历**（必做题）
（1）采用邻接矩阵存储图建图（包括有向图和无向图）
（2）实现图的深度优先和广度优先遍历
输入：图的顶点个数、顶点信息、边的数量、每条边的邻接顶点，如
4
A B C D　　//4 个顶点，分别是 A、B、C、D
3
A B
A C
B D　　//3 条边，分别是（A，B）、（A，C）、（B，D）
2、 **图的邻接表存储与遍历**（必做题）
（1）采用邻接表存储图建图（包括有向图和无向图）
（2）实现图的深度优先和广度优先遍历
（3）输入：图的顶点个数、顶点信息、边的数量、每条边的邻接顶点，如
3、 **图的应用**（选做题，程序设计竞赛的热门考点，不需要写实验报告）

二、程序设计的基本思想，原理和算法描述：

（包括程序的结构，数据结构，输入/输出设计，符号名说明等）

1. 邻接矩阵存储与遍历：
输入：顶点个数 n，顶点信息 vertex，边的数量 m，每条边的邻接顶点 edges
创建一个 n*n 的二维数组 matrix，所有元素初始化为 0
对于每条边（v1，v2）in edges：
　　找到 v1 和 v2 在 vertex 中的索引 i 和 j
　　matrix[i][j] = 1
　　如果是无向图，那么 matrix[j][i] = 1
深度优先遍历：
　　创建一个布尔数组 visited，所有元素初始化为 false
　　从第一个顶点开始，调用 DFS 函数
DFS 函数（顶点 v）：
　　标记 v 为已访问
　　对于 v 的每个邻接顶点 u：
　　　　如果 u 未被访问，那么调用 DFS 函数（u）
广度优先遍历：

创建一个布尔数组 visited，所有元素初始化为 false
创建一个队列 q
将第一个顶点加入 q，并标记为已访问
当 q 不为空时：
出队一个顶点 v
对于 v 的每个邻接顶点 u：
如果 u 未被访问，那么将 u 加入 q，并标记为已访问

2. 邻接表存储与遍历：

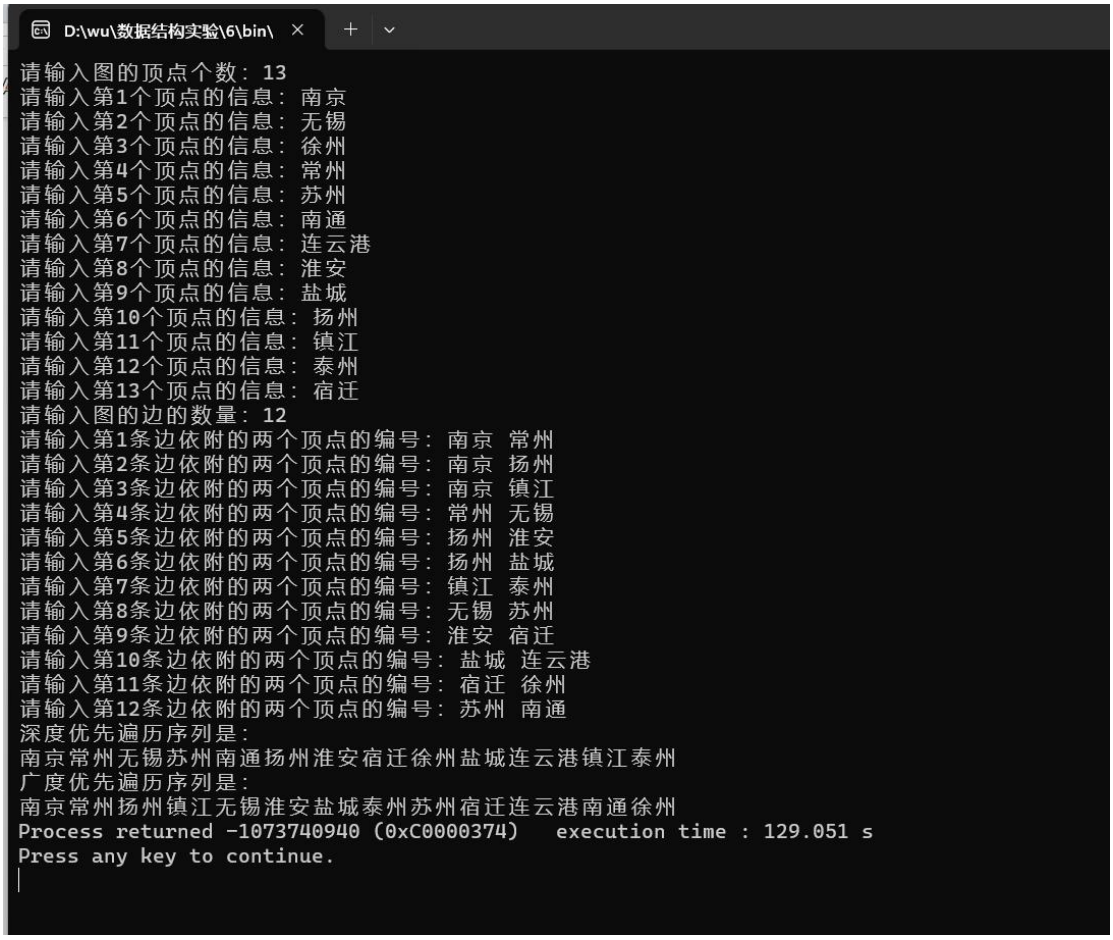
输入：顶点个数 n，顶点信息 vertex，边的数量 m，每条边的邻接顶点 edges
创建一个大小为 n 的数组 list，每个元素都是一个空列表
对于每条边（v1，v2）in edges：
找到 v1 和 v2 在 vertex 中的索引 i 和 j
将 j 添加到 list[i]中
如果是无向图，那么将 i 添加到 list[j]中
深度优先遍历：
创建一个布尔数组 visited，所有元素初始化为 false
从第一个顶点开始，调用 DFS 函数
DFS 函数（顶点 v）：
标记 v 为已访问
对于 v 在 list 中的每个邻接顶点 u：
如果 u 未被访问，那么调用 DFS 函数（u）
广度优先遍历：
创建一个布尔数组 visited，所有元素初始化为 false
创建一个队列 q
将第一个顶点加入 q，并标记为已访问
当 q 不为空时：
出队一个顶点 v
对于 v 在 list 中的每个邻接顶点 u：
如果 u 未被访问，那么将 u 加入 q，并标记为已访问

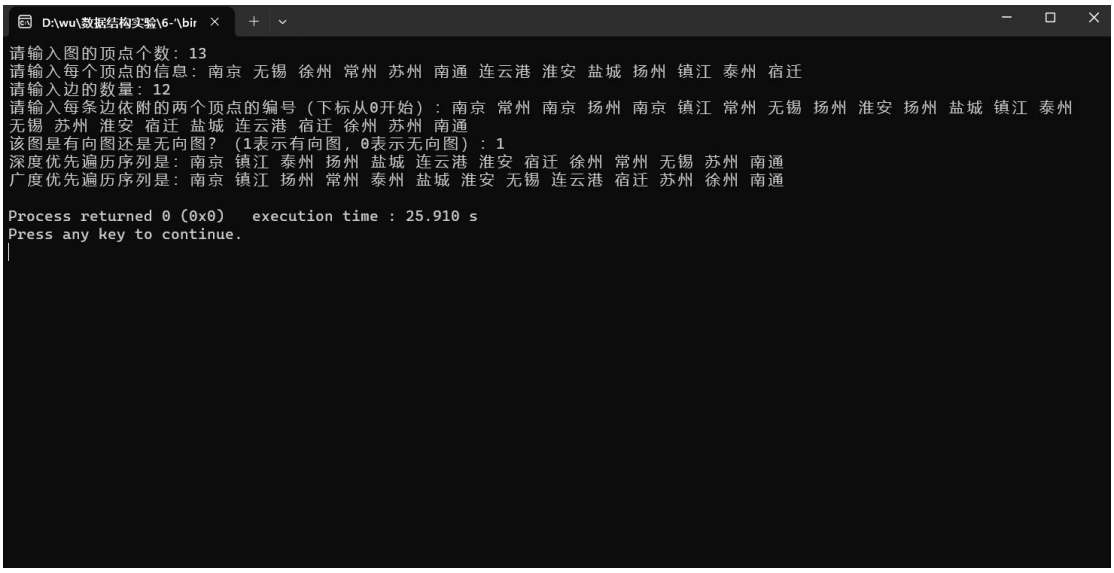
三、源程序及注释（说明每个文件的文件名即可，电子档的*.h 和*.cpp 文件压缩传到 FTP 上）：

实验 7.zip

四、运行输出结果：

（可以将运行结果抓图贴至此处）
输入以课上江苏十三地级市举例为例





五、调试和运行程序过程中产生的问题及采取的措施：

在实际编程过程中，可能会遇到各种问题，例如数组越界，空指针异常，递归深度过大等。对于这些问题，我们需要使用调试工具来定位问题，然后修改代码。例如，如果遇到数组越界，我们需要检查我们的索引是否正确；如果遇到空指针异常，我们需要检查我们是否正确初始化了所有的对象。

六、对算法的程序的讨论、分析，改进设想，其它经验教训：

这两个算法的时间复杂度都是 $O(n^2)$，其中 n 是顶点的数量。如果图的边数远小于顶点数的平方，那么使用邻接表可能会更有效率。此外，DFS 和 BFS 都有各自的用途，DFS 可以用来找到所有的连通分量，而 BFS 可以用来找到从一个顶点到另一个顶点的最短路径。

七、对数据结构教学的意见和建议：

暂时没有建议