|  |  |  |  |
| --- | --- | --- | --- |
| **《数据结构与算法》实验报告** | | | |
| **学生姓名** | 郭茁宁 | **院（系）** | 计算机科学与技术 |
| **学 号** | 1183710109 | **专 业** | 软件工程 |
| **实验时间** | 2019年12月6日（周五） | **实验地点** | 格物213室 |
| **实验项目** | **实验3/5：图型结构的建立与搜索（3学时）** | | |
| **实验目的：**将课程的基本原理、技术和方法与实际应用相结合，训练和提高学生组织、存储和处理信息的能力，以及复杂问题的数据结构设计能力和程序设计能力，培养软件设计与开发所需要的实践能力。  **实验要求：**灵活运用基本的数据结构和算法知识，对实际问题进行分析和抽象；结合程序设计的一般过程和方法为实际问题设计数据结构和有效算法；用高级语言对数据结构和算法进行编程实现、调试，测试其正确性和有效性。 | | | |
| **实验内容：**  **图的搜索（遍历）算法是图型结构相关算法的基础，本实验要求编写程序演示图两种典型存储结构的建立和搜索（遍历）过程。**   1. **分别实现图的邻接矩阵、邻接表存储结构的建立算法，分析和比较各建立算法的时间复杂度以及存储结构的空间占用情况；** 2. **实现图的邻接矩阵、邻接表两种存储结构的相互转换算法；** 3. **在上述两种存储结构上，分别实现图的深度优先搜索（递归和非递归)和广度优先搜索算法。并以适当的方式存储和显示相应的搜索结果（深度优先或广度优先生成森林（或生成树）、深度优先或广度优先序列和编号）；** 4. **分析搜索算法的时间复杂度；** 5. **以文件形式输入图的顶点和边，并显示相应的结果。要求顶点不少于10个，边不少于13个；** 6. **软件功能结构安排合理，界面友好，便于使用。** | | | |
| 数据结构定义：  class Point : 邻接表的点  class Point::Edge : 邻接表点所连的边  class Node : 邻接表点的链表元  class Queue : 队列（点）  class Stack : 栈（点） | | | |
| 算法设计与分析（要求画出核心内容的程序流程图）：   1. 邻接表的存储：     存储空间为O(N+M)，即边和点的总数   1. 邻接表🡪邻接矩阵   scanf("%d%d%d",&l,&r,&v);  matrix[l][r]/\*=matrix[r][l]\*/=v;    // 邻接表-->邻接矩阵     1. 深度优先搜索遍历   复杂度：邻接表-深搜-递归 O(n+2×m)  邻接矩阵-深搜-递归 O(n²)     1. 深度优先搜索（非递归）遍历   复杂度：邻接表-深搜-非递归 O(n+2×m)  邻接矩阵-深搜-非递归 O(n²)     1. 广度优先搜索遍历   复杂度：邻接表-广搜 O(n+2×m)  邻接矩阵-广搜 O(n²) | | | |
| 实验测试结果及结果分析：  Adjacency\_Matrix :  . 1 . . . . . . 1 . . .  . . 1 . . 1 . . . . . .  . . . 1 . . . . . . . .  . . . . 1 . . . . . . .  . . . . . . . 1 . . . 1  . . . . . . . . 1 . . .  . . 1 . . 1 . . . . . .  . . . 1 . . 1 . . . . .  . 1 . . . . . . . . . .  . . . . . 1 . . 1 . . .  . . . . . . 1 . . 1 . .  . . . . . . . 1 . . 1 .  Adjacency\_List :  a : ( a --> c ) ( a --> b )  b : ( b --> d ) ( b --> e )  e : ( e --> g )  g : ( g --> i )  i : ( i --> j ) ( i --> k )  d : ( d --> c )  f : ( f --> d ) ( f --> e )  k : ( k --> f ) ( k --> g )  c : ( c --> b )  l : ( l --> c ) ( l --> d )  h : ( h --> l ) ( h --> f )  j : ( j --> k ) ( j --> h )  List :  DFS\_recur a c b d e g i j k f h l  DFS\_non\_recur a b e g i k f j h l d c  BFS a c b d e g i j k h f l  Matrix :  DFS\_recur a b e g i k f d c j h l  DFS\_non\_recur a c b d e g i j h l f k  BFS a b c e d g i k j f h l    测试结果良好，显示界面排版清晰有条理，答案正确 | | | |
| 问题及解决方法：   1. 类的互相调用：点Point和边Edge类的成员变量都含有另一个的类，在不使用头文件的形况下无法调用，所以只能用类的嵌套，在调用Edge时使用Point::Edge； 2. 在搜索的过程中，要在入队/压栈/递归前就标记访问，若在出队/压栈/递归前时才标记，则会出现某些点多次遍历的情况； 3. 问题目标程序中许多的重复/相似操作，通过面向对象的操作可以将功能进行封装处理，节约代码量，提高主程序可读性； | | | |
| 源程序名称：lab3.cpp | | | |

注意：正文文字为宋体小4号，图中文字为宋体5号。行距为多倍行距1.25。

源程序与此报告打包提交，压缩包采用学号命名。