

**《数据结构》课程设计说明书**

**算法实现**

学 号： **2053182**

姓 名： **王润霖**

专 业： **信息安全**

题 号： **02**

二〇二二年八月

第一部分 题目

**1.题目描述**

以邻接矩阵的方式确定一个图，完成：

(1)建立并显示出它的邻接链表；

(2)以递归及非递归的方式进行深度优先遍历，显示遍历的结果，并随时显示栈的入、出情况；

(3)对该图进行广度优先遍历，显示遍历的结果，并随时显示队列的入、出情况。

程序采用标准交互式图形界面，需要有输入框、按钮、功能菜单等内容。

第二部分 软件功能

**2.1软件功能**

软件发布在：

<https://github.com/ChestnutSilver/DataStructureDesign>

(1)根据输入的邻接矩阵的相关信息，建立并显示出它的邻接链表；

(2)对该图进行深度优先遍历，显示深度优先遍历的结果；

(3)随时显示深度优先遍历中，栈的入、出情况；

(4)对该图进行广度优先遍历，显示广度优先遍历的结果；

(5)随时显示广度优先遍历中，栈的入、出情况；

**2.2软件样例**



第三部分 设计思想

**3.1课程设计相关背景**

**涉及数据结构：**图状结构。

**相关背景：图**是较线性表和树更复杂的、描述多对多关系的数据结构。图是用于表示对象之间联结关系的抽象数据结构，通常使用**顶点**（vertex）集和**边**（edge）集进行描述：顶点表示对象，边表示对象之间的关系。根据边是否有方向和权值，图又可细分为**有向图、无向图**以及**有向网、无向网**等。当边有方向时，一条边连接的两个顶点按方向分为原点（source vertex）和终点（destination vertex）。

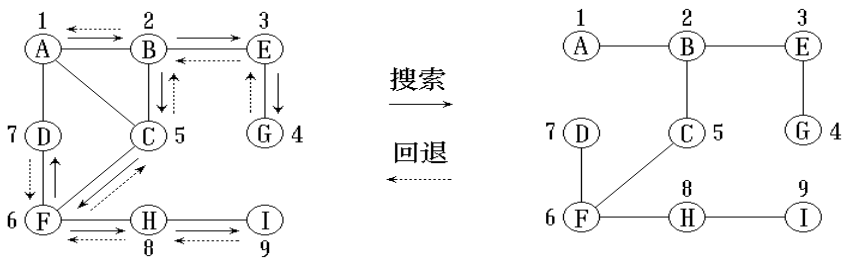
**3.2设计思想**

**（一）建立邻接表**

首先，输入总顶点数和总边数；然后建立顶点表：依次输入点的信息存入顶点表中，使每个表头结点的指针域初始化为NULL；最后创建邻接表，依次输入每条边依附的两个顶点，确定两个顶点的序号i和j，建立边结点，将此边结点分别插入到vi和vj对应的边链表的头部。

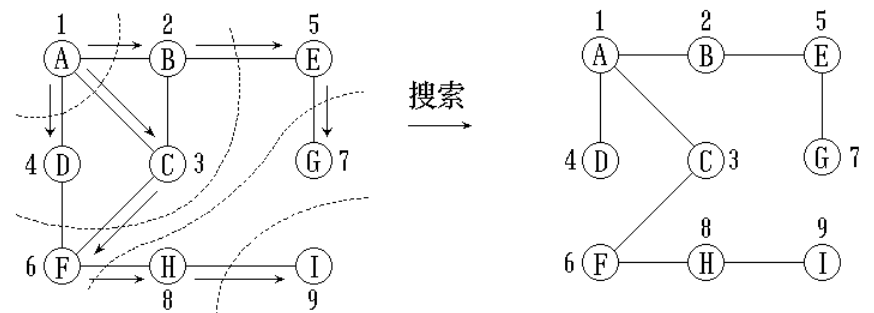
**（二）深度优先搜索（DFS）**

DFS算法：首先，DFS 在访问图中某一起始顶点 v 后，由 v 出发，访问它的任一邻接顶点 w1；再从 w1 出发，访问与 w1邻接但还没有访问过的顶点 w2；然后再从 w2 出发，进行类似的访问，… 如此进行下去，直至到达所有的邻接顶点都被访问过的顶点 u 为止。接着，退回一步，退到前一次刚访问过的顶点，看是否还有其它没有被访问的邻接顶点。如果有，则访问此顶点，之后再从此顶点出发，进行与前述类似的访问；如果没有，就再退回一步进行搜索。重复上述过程，直到连通图中所有顶点都被访问过为止。设计DFSTraverse函数：考虑到可能有非联通图，设计Traverse函数，使得所有的顶点都应该被访问过。（DFS搜索与回退过程如图所示）



**（三）广度优先搜索（BFS）**

BFS算法：在访问了起始顶点 v 之后，由 v 出发，依次访问 v 的各个未曾被访问过的邻接顶点 w1, w2, …, wt，然后再顺序访问w1, w2, …, wt 的所有还未被访问过的邻接顶点。再从这些访问过的顶点出发，再访问它们的所有还未被访问过的邻接顶点，如此进行，直到图中所有顶点都被访问到为止。设计BFSTraverse函数：考虑到可能有非联通图，设计Traverse函数，使得所有的顶点都应该被访问过。（BFS搜索与回退过程如图所示）



第四部分 逻辑结构与物理结构

**4.1逻辑结构**

**逻辑结构设计：图状结构**（无向图）

|  |
| --- |
| typedef int VertexType;  typedef struct ArcNode //表结点（边结点）  {  int adjvex; //该弧所指向的顶点的位置  struct ArcNode\* nextarc; //指向下一条弧的指针  int info;  }ArcNode; //边结点类型  typedef struct VNode //头结点  {  VertexType data; //顶点信息  ArcNode\* firstarc; //指向第一条依附该顶点的弧的指针  }VNode, AdjList[MAX\_VERTEX\_NUM];  typedef struct  {  AdjList vertices; //邻接表；存放各个顶点的数组  int vexnum, arcnum; //图的当前顶点数和边数  }ALGraph;  ALGraph G; |

**4.2物理结构**

物理结构描述：

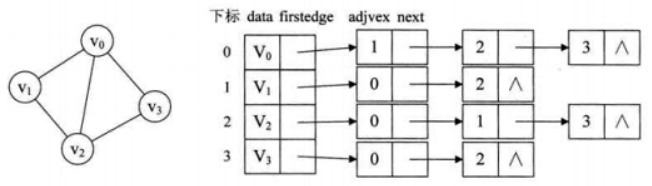
头结点的存储结构：

|  |  |  |
| --- | --- | --- |
| **VNode** | VertexType data; | ArcNode\* firstarc; |
| **头结点** | 4字节 | 4字节 |

表结点的存储结构：

|  |  |  |  |
| --- | --- | --- | --- |
| **ArcNode** | int adjvex; | struct ArcNode\* nextarc; | int info; |
| **表结点** | 4字节 | 4字节 | 4字节 |

无向图邻接表的存储表示：



第五部分 开发平台

**5.开发平台**

计算机型号 ：联想Legion Y7000 2020

处理器 Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz 2.50 GHz

机带 RAM 16.0 GB (15.9 GB 可用)

系统类型 64 位操作系统, 基于 x64 的处理器

操作系统 ：Windows 10 家庭版

开发语言 ：C++（C++11标准以上）

开发框架 ：QT

集成开发环境：Qt Version 5.12.10

第六部分 系统的运行结果分析说明

**6.系统运行结果分析说明**

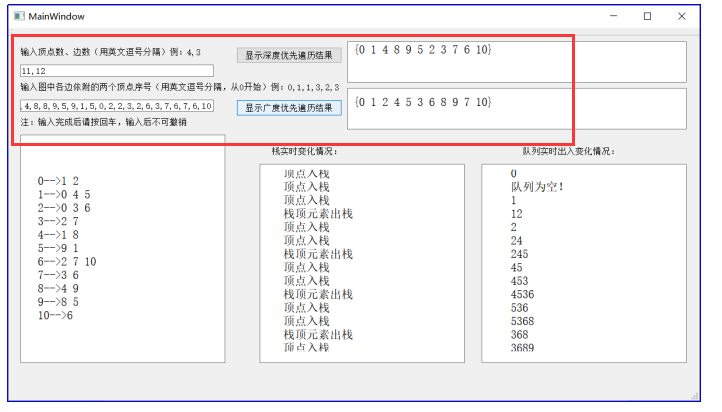
如左侧所示：

(1)根据输入的邻接矩阵的相关信息，建立并显示出它的邻接链表；

如右上所示：

(2)对该图进行深度优先遍历，显示深度优先遍历的结果；

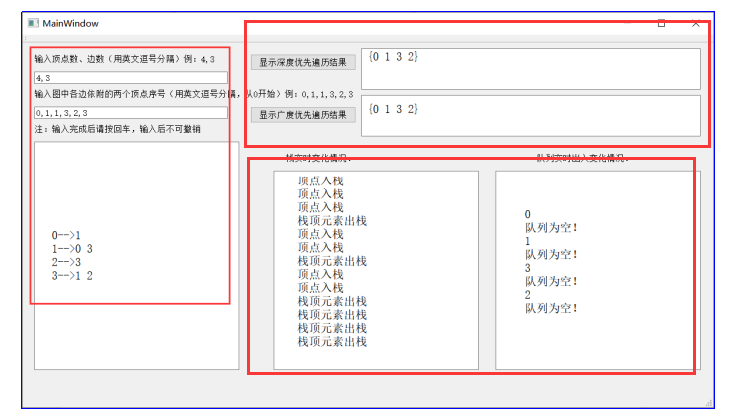
(3)对该图进行广度优先遍历，显示广度优先遍历的结果；



如右下所示：

(4)随时显示深度优先遍历中，栈的入、出情况；

(5)随时显示广度优先遍历中，栈的入、出情况；



第七部分 系统的安装与运行

**7.系统的安装与运行**

（一）打开Qt官网，找到下载目录Index of /archive/qt，选择需要下载的Qt版本，本软件版本为Qt5.12.10。



（二）选择安装组件，本软件安装组件为：



（三）已经附上所有程序执行需要的dll等文件，可以双击.exe文件运行。

第八部分 操作说明文档

程序操作简洁易懂，在操作界面已有说明：

（一）需要输入顶点数、边数，并用英文逗号分隔；

（二）需要输入图中各边依附的两个顶点的序号，注意从0开始标号，并用英文逗号分隔。

第九部分 参考文献

[1]霍亚飞.Qt Creator快速入门第二版[M].北京：北京航空航天大学出版社,2014-1

[2]严蔚敏，吴伟民.数据结构（C语言版）[M].北京：清华大学出版社,2007

[3]霍亚飞，程梁. QT5 编程入门[M]. 北京：北京航空航天大学出版社,2015-1