**作业 HW4 实验报告**

姓名：陆明奇 学号：2050283 日期：2021年11月23日

1. **涉及数据结构和相关背景**

图是一种较线性表和树更为复杂的数据结构。在线性表中，数据元素之间仅有线性关系，每个数据元素只有一个直接前驱和一个直接后继；在树形结构中，数据元素之间有着明显的层次关系，并且每一层上的数据元素可能和下一层中多个元素（即其孩子结点）相关，但只能和上一层中一个元素（即其双亲结点）相关；而在图形结构中，结点之间的关系可以是任意的，图中任意两个数据元素中间都可能相关。由此，图的应用极为广泛，已渗入到注=诸如语言学、逻辑学、物理、化学、计算机科学以及数学的其他分支中。

**2. 实验内容**

**2.1 图的存储结构**

**2.1.1 问题描述**

图是一种描述多对多关系的数据结构, 图中的数据元素称作顶点，具有关系的两个顶点形成的一个二元组称作边或弧，顶点的集合V和关系的集合R构成了图，记作G=（V,R）。图又分成有向图，无向图，有向网，无向网。图的常用存储结构有邻接矩阵、邻接表、十字链表、邻接多重表。图的基本操作包括图的创建、销毁、添加顶点、删除顶点、插入边、删除边、图的遍历。本题练习邻接矩阵和邻接表的创建。

**2.1.2 基本要求**

第1行输入一个数字1~4，1为有向图,2为有向网,3为无向图,4为无向网;

第2行输入2个整数n m，分别表示顶点数和边数，空格分割

第3行为n个字符的序列，一个字符表示一个顶点

后面m行，若前面选择的是图，则每行输入边的两个顶点字符，空格分割，若是网，则每行输入弧的两个顶点字符和弧的权值，空格分割

第1行输出顶点向量，顶点字符以一个空格分割

接着n行n列，输出邻接矩阵，每个数字占4位

接着n行，输出邻接表

**2.1.3 数据结构设计**

题目中要求输出图的邻接矩阵和邻接表，故采用邻接矩阵和邻接表两种结构来存储图

**2.1.4功能说明（函数、类）**

int LocateVex(MGraph G, char v)

功能：确定元素v在图的邻接矩阵存储结构中的位置

Para G：以邻接矩阵为存储结构的图

Para v：要查找的元素

返回值：元素在图中的位置

int LocateVex(ALGraph g, char v)

功能：确定元素v在图的邻接表存储结构中的位置

Para G：以邻接表为存储结构的图

Para v：要查找的元素

返回值：元素在图中的位置

bool CreateDG(MGraph& G, ALGraph& g)

功能：构造有向图

Para G：以邻接矩阵为存储结构的图

Para g：以邻接表为存储结构的图

返回值：构造是否成功

bool CreateDN(MGraph& G, ALGraph& g)

功能：构造有向网

Para G：以邻接矩阵为存储结构的图

Para g：以邻接表为存储结构的图

返回值：构造是否成功

bool CreateUDG(MGraph& G, ALGraph& g)

功能：构造无向图

Para G：以邻接矩阵为存储结构的图

Para g：以邻接表为存储结构的图

返回值：构造是否成功

bool CreateUDN(MGraph& G, ALGraph& g)

功能：构造无向网

Para G：以邻接矩阵为存储结构的图

Para g：以邻接表为存储结构的图

返回值：构造是否成功

bool CreateGraph(MGraph& G, ALGraph& g)

功能：按选择的序号构造图

Para G：以邻接矩阵为存储结构的图

Para g：以邻接表为存储结构的图

返回值：构造是否成功

void printVertex(MGraph G)

功能：输出顶点向量

Para G：以邻接矩阵为存储结构的图

返回值：无返回值

void printMGraph(MGraph G)

功能：输出图的邻接矩阵

Para G：以邻接矩阵为存储结构的图

返回值：无返回值

void printAlGraph1(ALGraph g)

功能：输出图的邻接表

Para G：以邻接表为存储结构的图

返回值：无返回值

void printAlGraph2(ALGraph g)

功能：输出网的邻接表（带权重）

Para G：以邻接表为存储结构的图

返回值：无返回值

**2.1.5 调试分析（遇到的问题和解决方法）**

这道题需要构造有向图，有向网，无向图，无向网四种图。同时每种图需要实现邻接矩阵和邻接表两种存储形式。构造这些图的函数名字和内容都差不多，写的时候容易分辨不清，造成错误。我将构造不同图的函数分开来编写并加入适当的注释以示区别，有效地避免了错误。此外，还需要注意每个结点插入邻接表的顺序，此题需要用头插法来插入每个结点。

**2.1.6 总结和体会**

通过这道题的练习，我掌握了有向图，无向图，有向网，无向网的邻接矩阵和邻接表的存储形式，熟悉了图的建立以及遍历等操作，

**2.2 图的遍历**

**2.2.1 问题描述**

本题给定一个无向图，用邻接表作存储结构，用dfs和bfs找出图的所有连通子集。所有顶点用0到n-1表示，搜索时总是从编号最小的顶点出发。使用邻接矩阵存储，或者邻接表（使用邻接表时需要使用尾插法）。

**2.2.2 基本要求**

第1行输入2个整数n m，分别表示顶点数和边数，空格分割。后面m行，每行输入边的两个顶点编号，空格分割。

第1行输出dfs的结果。第2行输出bfs的结果

**2.2.3 数据结构设计**

按照题目要求，此题采用邻接表这一存储结构来表示无向图。

**2.2.4功能说明（函数、类）**

bool CreateUDG(ALGraph& g)

功能：构造无向图

Para g：以邻接表为存储结构的图

返回值：构造是否成功

void DFS(ALGraph g, int v)

功能：深度优先搜索和v相联通的区域

Para g：无向图

Para v：深度优先搜索的起点

返回值：无返回值

void DFSTraverse(ALGraph g)

功能：深度优先搜素遍历无向图

Para g：无向图

返回值：无返回值

void BFS(ALGraph g, int v)

功能：广度优先搜索和v相联通的区域

Para g：无向图

Para v：广度优先搜索的起点

返回值：无返回值

void BFSTraverse(ALGraph g)

功能：广度优先搜素遍历无向图

Para g：无向图

返回值：无返回值

**2.2.5 调试分析（遇到的问题和解决方法）**

这道题需要采用尾插法来插入节点，使用头插法的话会导致答案错误。此外无论是用广度优先搜索还是深度优先搜索，都需要用到一个辅助数组来标记该点是否已被访问过，若不加这个辅助数组，会使程序陷入死循环。

**2.2.6 总结和体会**

通过这道题的练习，我掌握了用深度优先搜索和广度优先搜索两种不同方法来遍历无向图，理解了广度优先搜索和深度优先搜索两者的异同点。其中深度优先搜索可以运用栈这一数据结构辅助实现，而广度优先搜索可以用队列这一数据结构来辅助实现。

**2.3 关键路径**

**2.3.1 问题描述**

一个工程项目由一组子任务（或称活动）构成，子任务之间有的可以并行执行，有的必须在完成了其它一些子任务后才能执行，并且每个任务完成需要一定的时间。

对于一个工程，需要研究的问题是：

（1）由这样一组子任务描述的工程是否可行？

（2）若可行，计算完成整个工程需要的最短时间。

（3）这些任务中，哪些任务是关键活动（也就是必须按时完成的任务，否则整个项目就要延迟）。

现将这样一个工程项目用一个有向图表示，给定一组顶点，每个顶点表示任务之间的交接点（若任务2要在任务1完成后才可以开始，则这两任务之间必须有一个交接点，该点称作事件）。任务用有向边表示，边的起点是该任务可以开始执行的事件，终点是该任务已经完成的事件，边上的权值表示该任务完成需要执行的时间。

**2.3.2 基本要求**

输入：第一行包含两个整数n、m，其中n表示顶点数，m表示任务数。顶点按1~n编号。接下来m行表示m个任务，每行包含三个正整数ui、vi、wi，分别表示该任务开始和完成的顶点序号，及任务完成的时间。

如果该有向图有环，则工程不可行，输出0；否则第1行输出完成整个工程项目需要的时间，从第2行开始输出所有关键活动，每个关键活动占一行，按格式“ui->vi”输出，其中u和v为该任务开始和完成涉及的交接点编号。整数之间用空格表示。

**2.3.3 数据结构设计**

此题需要对所有结点进行拓扑排序，为了方便拓扑排序，我采用了邻接表的存储结构来表示有向网。

**2.3.4功能说明（函数、类）**

bool CreateDN(ALGraph& g)

功能：构造有向网

Para g：以邻接表为存储结构的图

返回值：构造是否成功

bool TopologiclaOrder(ALGraph& g, stack<VNode>& T)

功能：求各顶点事件的最早发生时间。若g无回路，则用栈T返回g的一个拓扑序列

Para g：以邻接表为存储结构的图

Para T：拓扑序列顶点栈

返回值：求拓扑序列是否成功

bool CriticalPath(ALGraph& g, stack<VNode>& T)

功能：输出g的各项关键活动

Para g：以邻接表为存储结构的图

Para T：拓扑序列顶点栈

返回值：输出g的各项关键活动是否成功

**2.3.5 调试分析（遇到的问题和解决方法）**

此题的输入数据中会有重复的边，第一次写的时候没有考虑到重复边，后一次输入相同的边会把该边之前的信息覆盖，导致错误。随后，我在每次输入时，首先判断该条边是不是重复边。若是重复边，则在所有重复边中选择权重最大的边，其余的相同边对求解没有影响，直接删去。

**2.3.6 总结和体会**

这道题使我熟悉了图的拓扑排序操作以及求图中关键路径的方法。其中，需要用一个栈来存取图经拓扑排序后的序列。在求vl时，由于逆拓扑排序必定在网中无环的前提下进行，亦可利用深度优先搜索来求取vl的值。

**2.4 最短路径**

**2.4.1 问题描述**

本题给出一张交通网络图，列出了各个城市之间的距离。请计算出从某一点出发到所有点的最短路径长度。

**2.4.2 基本要求**

输入：第一行包含三个整数n、m、s，分别表示n个顶点、m条无向边、出发点的编号。接下来m行，每行包含三个整数ui、vi、wi，分别表示第i条无向边的出发点、目标点和长度。

输出：一行，包含n个用空格分隔的整数，其中第i个整数表示从点s出发到点i的最短路径长度

**2.4.3 数据结构设计**

此题的数据范围为105，若采用邻接矩阵来存储无向网，则会超出内存限制，故采用邻接表这一存储结构来存储无向图

**2.4.4功能说明（函数、类）**

void ShortPath\_DIJ(int v0)

功能：求出从v0出发到图上其余各点的最短路径

Para v0：起始点

返回值：无返回值

**2.4.5 调试分析（遇到的问题和解决方法）**

一开始写的时候参考了书上的迪杰特斯拉算法来求出最短路径，采用的是邻接矩阵的存储形式，但最后两个测试点始终过不了。通过排查，发现最后两个测试点的数据大小超出了内存限制，从而导致了runtime error。随后我改用邻接表来存储图，其中，需要用到优先队列来快速求得一条权重最小的边，来提升时间效率。

**2.4.6 总结和体会**

此题使我熟悉了最短路径的实现方法，最主要的有迪杰特斯拉算法和弗洛伊德算法。迪杰特斯拉算法针对的是从某个源点到其余各顶点的最短路径，时间复杂度为O（n2）弗洛伊德算法针对的是每一对顶点之间的最短路径，时间复杂度为O（n3）。

**2.5 小世界现象**

**2.5.1 问题描述**

六度空间理论又称小世界理论。理论通俗地解释为：“你和世界上任何一个陌生人之间所间隔的人不会超过6个人，也就是说，最多通过五个人你就能够认识任何一个陌生人。“假如给你一个社交网络图，请你对每个节点计算符合“六度空间”理论的结点占结点总数的百分比。

**2.5.2 基本要求**

输入：第1行给出两个正整数，分别表示社交网络图的结点数N，边数M。随后的M行对应M条边，每行给出一对正整数，分别是该条边直接连通的两个结点的编号。

输出：对每个结点输出与该结点距离不超过6的结点数占结点总数的百分比。

**2.5.3 数据结构设计**

本题采用了一个vector<int>型的数组来表示每个结点及其对应的无向边，节约了空间，避免了建立邻接表的繁琐。

**2.5.4功能说明（函数、类）**

double BFS(int i)

功能：求与结点i距离不超过6的结点数占结点总数的百分比

Para i：起始点

返回值：该结点距离不超过6的结点数占结点总数的百分比

**2.5.5 调试分析（遇到的问题和解决方法）**

此题的输出要求是精确到小数点后两位，要特别注意，一开始写的时候没有注意到这一点导致输出全部错误。其次，广度优先搜索需要用到一个辅助数组来标记该点是否已被访问过，若不加这个辅助数组，会使程序陷入死循环。

**2.5.6 总结和体会**

通过这道题的练习，我掌握了用广度优先搜索从某一点来遍历无向图的操作，广度优先搜索可以用队列这一数据结构来辅助实现。此外，我还感受到了用vector来存储图的便利性，既节省空间，建立图的操作又不像邻接表那么复杂。

**3.实验总结**

通过这一章的习题，我熟悉了无向图，有向图，无向网，有向网的邻接矩阵和邻接表的存储形式，掌握了用深度优先搜索和广度优先搜索来遍历图和最小生成树的实现等。此外，我还掌握了求解关键路径和最短路径的一系列算法，如拓扑排序，迪杰特斯拉算法和弗洛伊德算法等。