**数据结构**

2022

**实 验 报 告**

|  |  |
| --- | --- |
| 实验项目名称： | 数据结构实验八 |
| 班级： | 21级9班 |
| 学号 | 2021302121185 |
| 姓名： | 陈聪睿 |
| 指导教师： | 董红斌 |
| 实验时间： | 2022.5.12 |

**实验一： 实现图的邻接矩阵和邻接表存储**

**一、实验要求**



**二、实验环境**

**Visual studio 2019/ Windows平台**

**三、实验步骤及思路**

（1）题目分析与主要步骤

根据题目要求可知，需要构建一个邻接矩阵结构和一个邻接表结构，以及若干实现其存储结构的基本功能的方法。

1. 邻接矩阵

邻接矩阵的构建，可以使用一个二维数组存储。

同样的，我们首先使用一个二维数组将原有数据存储起来，尔后用我们编写的CreateMat函数，将非空（不是INF）位置的数据存储进入该表对应位置中；将空（INF）的位置的数据存储为∞，加以区分。

同时，我们可以编写DispMat函数，使用双循环遍历的方法遍历整个二阶矩阵并输出，实现邻接矩阵的输出。

1. 邻接表

相比于邻接表，其构造方法稍微复杂些。

它的存储结构可分为以下三部分：表头结点ArcNode（表示顶点信息，其存储着顶点的信息以及其指向的第一条边）、边界点VNode（表示边各个边的信息，其存储着该边的终点编号，下一条边的地址以及其权值等信息）、一个结构体（其存储着由头结点构建起来的一维数组，用于查询每个顶点对应信息。以及图中顶点数n和边数e）。

而其实现方法包括创建邻接表CreateAdj函数、输出邻接表函数DispAdj、销毁邻接表函数DestroyAdj。

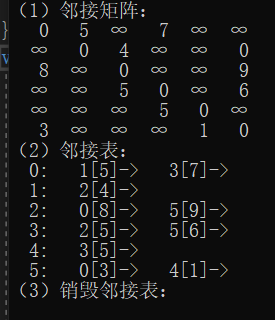
关于创建邻接表，我们可以首先借由邻接矩阵创建。通过二重循环遍历整个邻接矩阵，并将查询A[i][j]是否有数据存在，将其按照邻接矩阵的对应关系，将其存入新建结点，并插入其对应位置中。

输出邻接表DispAdj较为简单，即直接遍历整个邻接表，并按照顺序输出。

销毁邻接表DestroyAdj与遍历类似，只不过是直接将其释放掉。

（2）实验具体步骤截图

测试数据：题目已给出，结果如下图。



**四、实验结果及分析**

可以看出已经按照需要打印出了正确结果。

**五、总结**

可以看出，有向图的邻接矩阵和邻接表有如下特点：

有向图的邻接矩阵有如下性质：

有向图的邻接矩阵可能是不对称的；

顶点出度=第i行元素之和；

顶点入度=第i列元素之和；

顶点的度=第i行元素之和+第i列元素之和

一个图的邻接矩阵表示是唯一的。

特别适合于稠密图的存储。

邻接表的特点如下：

邻接表表示不唯一。

在有向图中，我们只记录出度边。存储空间为O（n+2e），适合于稀疏图。

**实验二： 采用普里姆算法求最小生成树**

**一、实验要求**



****

**二、实验环境**

**Visual studio 2019/ Windows平台**

**三、实验步骤及思路**

Prim算法的算法流程如下：

（1）初始化U={v}。v到其他顶点的所有边为候选边；

（2）重复以下步骤n-1次，使得其他n-1个顶点被加入到U中：

从候选边中挑选权值最小的边输出，设该边在V-U中的顶点是k，将k加入U中；

考察当前V-U中的所有顶点j，修改候选边：若(j，k)的权值小于原来和顶点k关联的候选边，则用(k，j)取代后者作为候选边。

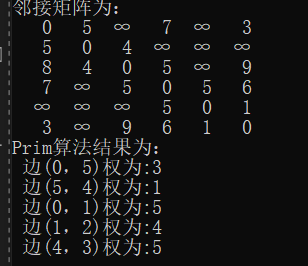
根据上述原理，我们可以只考虑V-U集合的顶点，建立两个数组closest和lowcost来存储V-U中顶点j到U中顶点的最小边。

对于v - u 中的一个顶点j' 它的最小边对应U中的某个顶点，用closest[j] 保存U中的这个顶点。 顶点j的最小边对应U 中的顶点k, 有closest[j]= k, 并且用lowcost[j]存储该最小边的权。也就是说，这样的最小边为（closest [j], j)边，对应的权为lowcost[j]。

为了判断一个一个顶点属于U 集合还是属于V-U 集合，若某个顶点i 有lowcost[i] = 0, 表示i属于U; 若O < lowcost[i] <∞，（或者lowcost[i]！=0), 表示i不属于V-U。

（2）实验具体步骤截图

测试数据：题目已给出，结果如下图。



**四、实验结果及分析**

可以看出已经按照需要打印出了正确结果。

**五、总结**

通过本实验，可以知道Prim算法时间复杂度为O（n2），与边数e无关，故适用于稠密图求最小生成树。