

## 实验 5 日志

### 一. 实验设计

实验 5 要求我们用邻接矩阵和邻接表两种方式实现图的抽象数据类型 ADT。邻接矩阵本质上来说是可以看做是一个二维数组，我们可以利用指针的指针来实现一个二维数组，存储图的各个顶点之间的关系。

邻接表从结构上来说也是一种二维数组，但每个行数组的长度都是动态可变的，其是根据顶点之间的关系来确定可变动态数组的长度的，实现上，我们可以用一个结构体数组来存储图的各点之间的关系和边上权值，与每个顶点连接的点可以用一个链表来存储，链表的每一个结点为了方便，可以定义为一个边类的对象，其中包含对应边的序号和相应的权值。

### 二. 实验记录

2019 年 11 月 21 日

用邻接矩阵来实现图，关键在于对应好每个点和二维数组下标的关系，这样我们可以用一个数组来存储顶点的信息，写一个函数可以通过顶点的值去返回对应序号值，实现了点和二维数组下标的对应关系。

怎样将图上边的连接关系和二维矩阵联系起来：对于邻接矩阵，按照行列来看如果对应序号有边相连，那对应应在矩阵中的位置为 1，否则为 0。对于邻接表，将每个与结构体数组中顶点相关联的点都一次存入链表中，包括该点的序号和两点之间的权值。

如何从邻接矩阵中求出每个顶点的出度和：将每个顶点对应的矩阵中每行的值相加，得到的就是该顶点的出度和，求出这些出度和中的最大值即可。对于邻接表实现的图，每个顶点的出度就是该顶点对应链表的长度，可以利用链表中的函数 `length()` 求出出度和。

2019 年 11 月 22 日

用邻接表来实现图，可以在一定程度上减少空间的消耗（对于某些图）。首先需要有一个结构体数组来存储各个顶点，以及这些点和与他们相关的点的存储，前者用一个变量来存，后者用一个链表来存。在实现上，不仅需要图的 ADT 和相应的操作函数，还需要链表的 ADT 以及操作的实现。

如何利用邻接表输出对应图的邻接矩阵：可以建立一个二维数组，整个数组赋初值为 0，然后先遍历每一个点，遍历到某个点的时候，从它的链表的头结点开始，依次遍历整个链表，将两个点和对应权值用函数在对应二维数组的位置上赋给相应的值。

## 二. 心得和问题

2019 年 11 月 22-23 日

1. 指针使用上的问题，在邻接表实现图上用到了链表，要注意用链表来处理问题时，要注意指针的初始化，不然在后续处理上可能对某些操作存在影响，造成程序异常出错。
2. 在输入每一个顶点信息后，要注意将每一个顶点标记为已有，在后面输入点的输出可以避免重复。另外，可以根据邻接矩阵是否是对称矩阵来判断图是有向图还是无向图。