# 二、线性表

1.顺序存储结构：

初始化：(List*)malloc(sizeof(List))开辟空间;Ptrl->last=-1；

指定位置插入O(n)：从last往左到i，依次往右挪

指定位置删除O(n)：从i往右到last，依次往左挪

2.链式存储实现：不要求逻辑相邻元素物理上也相邻，插入删除不需移动元素，只需修改链

求表长O(n)：从头指针开始遍历，每挪一位num++

查找O(n):按序号找/按值找

插入O(n)：先查找，s=(List*)malloc(sizeof(List)),在改链表指针

删除O(n)：先查找，在改链表指针；free(s)释放空间

3.广义表Generalized List（二元多项式）：系数也可以指向另一个一元多项式（用tab）区分data类型,用union{}包起来

4.多重链表：节点的指针域有多个（双向链表不是多重链表）----树，图

4.1.十字链表：存储稀疏矩阵（里面0很多）

结构：head（行列头）+term（非零元素）+入口term

行指针Right

列指针Down

数据value 行row 列col

【图】

5.堆栈Stack：只在栈顶Top做插入删除

5.1.表达式:运算数+运算符号

中缀表达式：a+b*c-d/e

后缀表达式：abc*+de/- 从左往右扫描，记住数，遇到符号处理最后两个数（后放进去的先拿出来） O（n）

5.2.插入数据：入栈push

删除数据：出栈pop

后入先出：LIFO

5.3.栈的顺序存储：一维数组+记录栈顶元素位置的int变量

top=-1代表栈空

5.4.栈的链式存储：

一个单链表（只能前找后，后找不到前）（链栈）

top必须在链表的头，不能在尾（删除后找不到前一个数据）

5.5.堆栈应用：

中缀表达式->后缀表达式【图】

函数调用及递归

深度优先搜索

回溯算法

6.队列Queue：只能在一端插入，另一端删除

6.1.操作：

插入：入队列addQ：rear+1

删除：出队列DeleteQ：front+1

先进先出：FIFO

6.2.顺序存储实现：

6.2.1.结构：

一维数组+

记录头元素位置int变量front：指示队列第一个元素的前一个元素

尾rear：指示队列最后一个元素

空队列：front=rear=-1

6.2.2循环队列

问题：front=rear无法判断空或满

解决：使用size或tag作为额外标记

仅使用n-1个元素，不把数组放满

6.3.链式存储实现LinkQueue：单链表

链表头对应front：做删除

链表尾对应rear：做添加

空队列：front=null

7.应用：多项式加法运算