

线性表

二元组(K, R) $K=\{a_0, \dots, a_{n-1}\}$ $R=\{r: \text{线性关系}\}$

数学定义:唯一的开始结点(没有前驱,但有唯一的直接后继),唯一的终止结点(没有后继,但有唯一的直接前驱),内部结点(有唯一的直接前驱,也有唯一的直接后继),线性表的长度(包含的结点个数,为零称为空表), r 是前驱关系(应具有反对称性和传递性)

要求:内部结点具有相同的数据类型,每一个元素都有自己的位置

运算分类:构造函数,析构函数,位置 \leftrightarrow 内容,增删改,辅助性操作(游标,当前长度)

定长,静态的存储结构(向量式的一维数组)地址相邻表达线性关系,长度固定

变长,动态的存储结构(链式存储结构)指针指向表达线性关系

顺序表(向量)

采用定长的一维数组存储结构

元素类型相同,存储在连续的空间中,元素索引值唯一,读写元素方便,使用常数作为向量长度,程序运行时不变

类定义:保存最大长度,当前长度,当前处理位置

查找元素:位置 \rightarrow 值 $O(1)$ 值 \rightarrow 位置 $O(n)$

插入元素:插入位置是否越界($currlen$) \rightarrow 当前长度是否溢出($maxsize$) \rightarrow 先移动再插入 $O(n)$

删除运算:删除位置是否有效 \rightarrow 是否向下溢出 \rightarrow 元素向前移动 $O(n)$

链表

指针指向保持前驱关系,节点不必物理相邻,长度动态变化,单链表/双链表/循环链表

引入头结点(哨兵节点),被 $head$ 指向

链表检索:-1 定位到头结点,位置从零开始计数,返回空或者指向 i 的指针

链表插入:插到第 i 个 \rightarrow 去找 $i-1$ 的指针(它是空则插入位置无效),如果插到最后($next = NULL$)维护尾指针

链表删除:找 $i-1$ 的指针(不能为空/尾节点),删除

头结点的好处:链表第一个位置的操作和其他位置的操作一致,空表和非空表的处理统一

单链表的不足:不能有效找到前驱 \rightarrow 引入双链表

双链表的插入与删除:都要注意边界条件的判断,和操作的次序

循环链表:链接单/双链表的头尾

顺序表的优点:没有指针,没有额外开销(结构性开销),访问非常便利,不适宜经常插入删除,考虑最大长度

链表的优点:无需知道长度,允许长度动态变化,适应经常插入元素的情况,不适宜经常读的情况,指针存储比例大也不合适

顺序表适合静态数据,链表适合动态数据