

数据结构

变量

存储结构

基于地址 一一对应

单数据到连续数据

顺序表

存储结构

基于地址连续

数组名与数组元素访问关系

$&num[x] = num + x$

存储特性

插入效率

指定位置

$O(n)$

删除效率

指定位置

$O(n)$

查询效率

指定位置

$O(1)$

区间移动效率

$O(n)$

地址分配

动态

静态

优缺点

优点

访问下标  $O(1)$

缺点

由于保证地址连续 插入/删除/移动  $O(n)$

存储结构

地址未必连续的内存块

哪有分哪郑爽定式

前后序问题

全体样本空间不保证 前后序相邻

前序  $O(n)$

链表

集合需求  
由于顺序表 插入 删除 区间移动的缺点提出链表

单链表

放弃前序 记录首元素与后序或前序

特点

前序  $O(1)$

后序  $O(n)$

放弃前序节省一半空间

存储特性

双链表

前后序都记录

特点

前序  $O(1)$

后序  $O(1)$

删除元素很方便找到前后序

循环双链表

尾结点next与头结点pre不再为空

特点

头结点  $O(1)$

尾结点  $O(1)$

遍历结束特点  $next = head$

建表法

头插法

逆序

尾插法

正序

优缺点

优点

插入

$O(1)$

删除

$O(1)$

缺点

访问下标

$O(n)$

疑难

链表大部分问题是由于指针理解导致的

指针只是做一个指向 并不会影响指向的内存!!!

指针只是访问内存的工具 只有通过其访问后才能去修改 如  $p \rightarrow member$ .....

因此  $p \rightarrow next = null$   $p \rightarrow next = q$  是去修改指针指向内存的next成员指针的指向  $p = p \rightarrow next$ 而这里是修改指针的指向

线性表

操作

基础

查询元素

链表  $O(n)$

顺序表  $O(n)$

删除元素

链表  $O(1)$

顺序表  $O(n)$

优化 打标记  $O(1)$  但转化为了链表

访问下标

链表  $O(n)$

顺序表  $O(1)$

区间翻转

链表

头插法

链表最特别的逆序算法

递归回溯

空间换时间

存储到其他结构转换成为其他结构问题 比如顺序表

顺序表

关系映射

双指针

空间换脑子

逆运算性质

$A^{-1} \cdot A^{-1} = A$

变形题型

前p个交换到后面 保持序列不变

顺序表

朴素解

空间换时间

时间换空间

逆运算特性

整体逆运算 再区间逆运算

特殊解法

空间换时间以及异或

链表

查询  $O(n)$  修改  $O(1)$

单调问题

具体问题根据单调性质去贪心

特点

末尾增删  $O(1)$

查找  $O(n)$