数据结构和算法

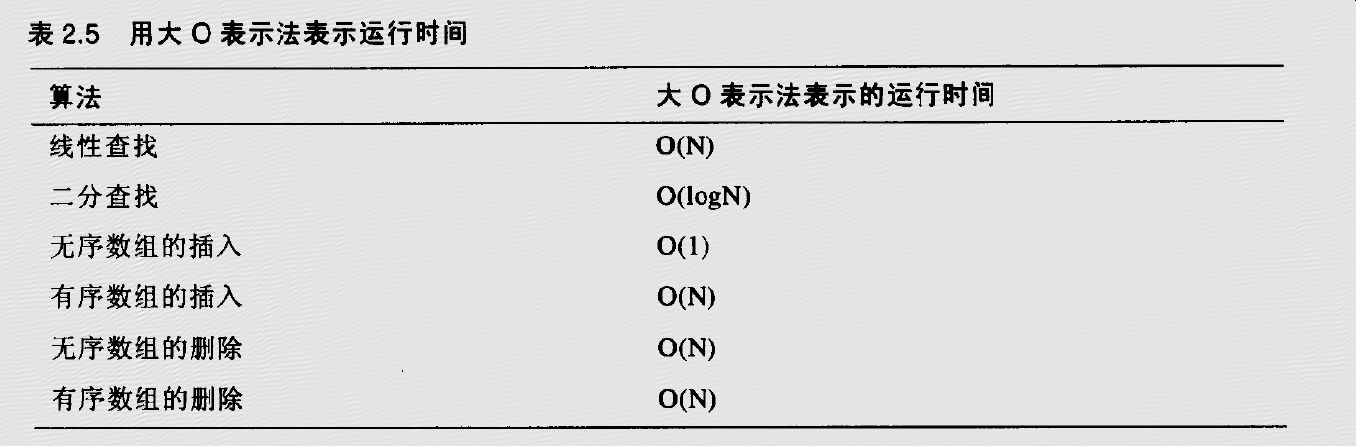
1. **大O表示法**

时间复杂度：

1.无序数组的插入：常数

2.线性查找：与数组的长度N成正比

3.二分查找：与log(N)成正比



**2.线性表**

顺序存储结构：数组

插入，删除 复杂度为O(n)，

读取 O(1)

链式存储结构

单向链表：每个元素存储下一个元素的信息，只有后继

循环链表：最后一个元素的后继指向了第一个元素

静态链表：c语言中

双向链表：每个元素存储前驱和后继，前驱指向上一个元素，后继指向下一个元素

**3. 栈和队列**

一、栈：

栈其实就是线性表的一个特例，后进先出LIFO。区分栈底和栈顶。分为顺序存储结构和链式存储结构

1.两栈共享存储空间：

定义一个数组，一个栈以0开始，另一个栈以n-1开始，这样最大化利用空间但是要注意需要用同样的数据类型

2.顺序栈和链式栈，不管是进栈还是出栈，时间复杂度都是O(1)。如果栈在使用过程中大小未知，最好使用链式，因为链式栈几乎不存在溢出情况。如果栈的变化已知，使用顺序栈比较好，因为是连续的存储空间。

3.栈的应用：递归

注意：每个递归至少有一个条件，满足时就退出，否则会陷入无限循环

递归会建立函数的副本，相对于直接迭代循环，会耗费更多的内存和时间

4.栈的应用：四则运算表达式

9 + ( 3 – 1 ) \* 3 + 10 / 2 9 3 1 – 3 \* + 10 2 / +

中缀表达式转后缀表达式：数字进栈，遇到符号就计算之前两位数字出栈计算结果然后再进栈

后缀表达式转中缀表达式：符号进栈，遇到优先级低于栈顶的，栈内符号出栈

二、队列

队列是一种先进先出FIFO的线性表。插入的一端称为队尾，删除的一端称为队头。

队列也分为顺序存储和链式存储

1.顺序存储

插入操作就是在队尾增加，时间复杂度为O(1)；

删除操作需要队头删除，后面元素向前移动，所以时间复杂度为O(n)。

包含两个指针，front指针指向队头，rear指针指向队尾元素的下一个位置。Front和rear相等的时候，说明这个队列是空队列

2.循环队列：

队列头尾相接的顺序存储结构称为循环队列，插入和删除的时间复杂度都为O(1)

计算队列长度公式：(rear-front+queenSize)%queenSize

3.链式存储结构队列

插入和删除的时间复杂度都为O(1)

确定队列最大值情况，用循环队列。不确定用链式队列

**4.树**

1.判断一个数是不是完全二叉树的方法：

按照满二叉树排序，如果中间有不连续的数字，就不是完全二叉树

2.二叉树性质

1.在二叉树的第n层，最多有2n-1个节点

2.深度为k的二叉树，最多有2k-1个节点

3.任意一个二叉树，终端节点数n0，度为2的节点数为n2，那么n0=n2+1

N = n0+n1+n2

4.具有n个节点的完全二叉树的深度为[log2n]+1

5.