

一、二叉搜索树（二叉排序树）

1.1 定义：二叉树且每个节点的左子树的所有节点的数值比根节点的小，右子树的所有节点的数值比根节点的大

2.1 二叉搜索树操作的特别函数

- (1) 查找元素x，返回所在节点的地址
- (2) 查找最小元素并返回所在节点的地址
- (3) 查找最大元素并返回所在节点的地址
- (4) 二叉搜索树的插入算法
- (5) 二叉搜索树的删除算法
- (6) 二叉搜索树的建立--实际上是逐个插入

2.3 含n个元素的二叉搜索树的高度范围在n到 $\log n$ 之间

二、平衡二叉树（不一定是二叉排序树）

why? 若二叉排序树插入结点是序列是递增或者递减的，二叉排序树的高度显著增加，退化为单链表，查找效率显著降低，为了避免二叉排序树左右子树高度相差太大，引入了平衡二叉树的概念

2.1 平衡因子

左右子树的高度差

2.2 平衡二叉树的定义

(1) 空树

(2) 树不空，但任意节点的左右子树的高度差不超过一

2.3 平衡二叉树的节点数与树高的关系

给定节点数为 n 的平衡二叉树的最大高度为 $O(\log n)$

2.4 平衡二叉树的调整

(1) RR旋转：破坏节点插在了被破坏节点的右子树的右子树上，需要将左子树根节点右旋

(2) LL旋转：：破坏节点插在了被破坏节点的左子树的左子树上，需要将右子树根节点左旋

(3) LR旋转：：破坏节点插在了被破坏节点的左子树的右子树上，需要将左子树的右子树的根节点先左旋再右旋

(4) RL旋转：：破坏节点插在了被破坏节点的右子树的左子树上，需要将右子树的左子树的根节点先右旋再左旋

三、最优二叉树（哈夫曼树）

3.1 如何构建哈夫曼树

利用最小堆的结构，堆中数组是个结构体数组，数组元素为哈夫曼树的结构体

3.2 哈夫曼树的特点

(1) 没有1度顶点

(2) n 个叶节点的树共有 $2n-1$ 个节点

(3) 左右子树还是哈夫曼树

四、线索二叉树

why?利用二叉链表的空指针域（如果指向了孩子结点的指针就不动它，如果要找它的中序前驱就找它左子树最右结点，同理找它的中序后继就找它右子树最左结点） 指向

遍历的直接前驱和直接后继，使得二叉树的遍历变得如链表的遍历一般容易

how?通过中序遍历构造中序线索二叉树

can do what?找到中序遍历的第一个结点和最后一个结点；

给定一个结点的指针，找到它的直接后继和直接前驱；

找到中序遍历的一个结点，然后依次找结点的后继实现中序遍历