**数据结构与算法**

# 树形结构

## 二叉树

### 定义

### 五种基本形态

### 性质

1. 二叉树的第层上最多有个节点。
2. 高度为的二叉树中，最多有个节点。
3. 在任意一棵二叉树中，如果叶子节点的个数为，度为2的节点数为，则。
4. 二叉树的子树有左右之分，顺序不能颠倒。
5. 若采用连续储存的方式存放二叉树，则节点下标之间的关系：
   * 1. 若某个节点的下标为 i ，则这个节点的父节点的下标为 i / 2。
     2. 若某个节点下标为 i ，且节点的度为2，则这个节点的左子节点的下标为 2 \* i + 1 ，右子节点的下标为 2 \* i + 2 。

## 完全二叉树

## 满二叉树

## 二叉排序树（Binary Sort Tree，BST）（二叉搜索树，二叉查找树）

### 定义：

它或者是一棵空树，或者是具有下列性质的二叉树： 若它的左子树不空，则左子树上所有结点的值均小于它的根结点的值； 若它的右子树不空，则右子树上所有结点的值均大于它的根结点的值；它的左、右子树也分别为二叉排序树。

## 平衡二叉树（AVL）

### 定义：

平衡二叉树是一种二叉排序树，它或者是一棵空树，或者它的左右两个子树的高度差的绝对值不超过1，并且左右两个子树都是一棵平衡二叉树。

### 构造一棵平衡二叉树

动态地调整二叉排序树平衡的方法为：每插入一个结点后，首先检查是否破坏了树的平衡性，如果因插入结点而破坏了二叉查找树的平衡，则找出离插入点最近的不平衡结点，然后将该不平衡结点为根的子树进行旋转操作，我们称该不平衡结点为旋转根，以该旋转根为根的子树称为最小不平衡子树。

#### LL旋转

插入结点的左孩子的左子树而导致失衡的情况需要进行LL旋转。

#### RR旋转

插入结点的左孩子的左子树而导致失衡的情况需要进行RR旋转。

#### LR旋转

插入旋转根的左孩子的右子树而导致失衡的情况需要进行LR旋转。

#### RL旋转

插入旋转根的右孩子的左子树而导致失衡的情况需要进行RL旋转。

### 如何回溯修改祖先结点的平衡因子

* 只有根结点到插入结点路径（称为插入路径）上的结点的BF值会被改变。
* 当一个结点插入到某个结点的左子树时，该结点的BF值加1；当一个结点插入到某个结点的右子树时，该结点的BF值减1。

### 适用情况

AVL树适合用于插入与删除次数比较少，但查找多的情况

## 红黑树

### 性质

1. 每个节点要么是红色，要么是黑色。
2. 根节点永远是黑色的。
3. 所有的叶节点都是空节点（即 null），并且是黑色的。
4. 每个红色节点的两个子节点都是黑色。（从每个叶子到根的路径上不会有两个连续的红色节点）
5. 从任一节点到其子树中每个叶子节点的路径都包含相同数量的黑色节点。

## 哈夫曼树