

## 2020/06/08\_数据结构\_第7课\_二叉平衡树

笔记本： 数据结构

创建时间： 2020/6/8 星期一 5:47

作者： ileemi

标签： 二叉平衡树

---

- [二叉搜索树](#)
- [二叉平衡树](#)
  - [二叉平衡树的实现原理](#)
  - [平衡二叉树的失衡调整](#)
  - [左单旋](#)
  - [右单旋](#)

## 二叉搜索树

各个操作的时间复杂度：

- 插入 --> 对数阶 ( $\log(n)$ )
- 删除 --> 对数阶 ( $\log(n)$ )
- 查询 --> 对数阶 ( $\log(n)$ )
- 修改 --> (先修改再插入  $2\log(n)$ ) 对数阶 ( $\log(n)$ )

二叉搜索树的时间复杂度不稳定，当出现左斜，右斜树时，时间复杂度和链表一样。

## 二叉平衡树

为了避免二叉搜索树变成 "链表"，出现了平衡二叉树，让树的结构看起来尽量 "均衡"，左右子树的结点数量尽量一样多。

性质：

- 可以是空树
- 不是空树时，任何一个结点的左子树与右子树都是平衡二叉树，树中的任意结点，其左子树和右子树的高度差的绝对值不超过1。

## 二叉平衡树的实现原理

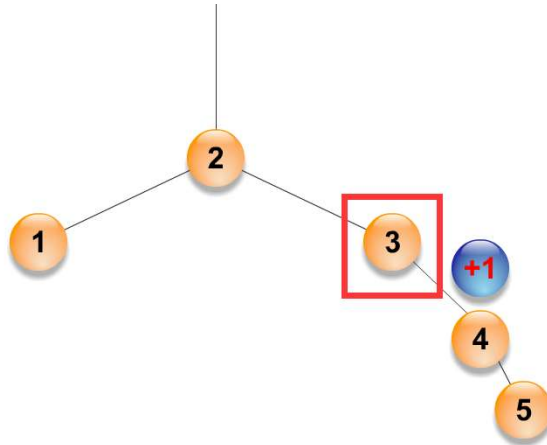
二叉树平衡构建的基本思想就是在构建二叉排序树的过程中，每当插入一个结点时，先检查是否因插入而破坏了树的平衡性，若破坏了树的平衡性，需要找出最小不平衡树（距离插入点最近的，且平衡因子（左子树的高度减去右子树的高度）的绝对值大于1的结点为根的子树）。在保持二叉排序树特性的前提下，调整最小不平衡树中各个结点之间的链接关系，进行相应的旋转，使之成为新的平衡子树。

## 平衡二叉树的失衡调整

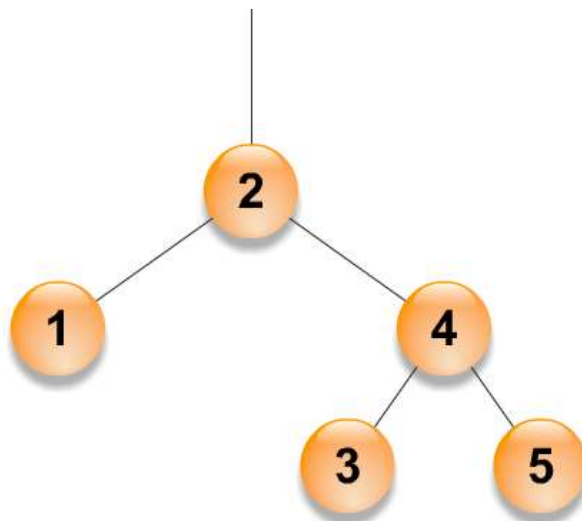
平衡二叉树的失衡调整主要是通过旋转最小不平衡树来实现的。

**最小不平衡树：** 在新插入的结点向上查找，以第一个平衡因子的绝对值超过1的结点为根的子树称为最小不平衡子树。

树的失衡，可能是由多棵子树同时失衡的，当出现这种情况，只需要调整最小不平衡子树，就可以将不平衡的树调整为平衡的树。



如上图，根结点 2 的平衡因子为：2，根结点2的右孩子 3 的平衡因子为：2，这个时候，整棵树同时存在两棵不平衡子树，这里以 3 为根的子树就是最小不平衡子树，此时只需要将最小不平衡子树进行左旋，这个整数就可以恢复平衡。



## 左单旋

左孩子的高度减去右孩子的高度差值小于-1时，进行左旋转。

左单旋，将不平衡的节点从右往左拉，其不平衡结点的右孩子变成了父节点，并把其右孩子的左节点出让给降级后节点的右子节点。

## 右单旋

右单旋，将不平衡的节点从左往右拉，其不平衡结点的左孩子变成了父节点，并把其左孩子右左节点出让给降级后节点的左子节点。

左旋就是往左变换，右旋就是往右变换。不管是左旋还是右旋，旋转的目的都是将节点多的一支 出让节点 给另一个节点少的一支。