

【数据结构】Day7

▼ Class	Advanced Data Structures
📅 Date	@December 12, 2021
🔗 Material	
# Series Number	
☰ Summary	AVL树双旋

【Ch4】树

4.4.2 双旋转

单旋转对于情形2和3，及插入左（右）子树的右（左）孩子后树不平衡，**无效**。

问题在于子树Y太深，单旋转没有降低它的深度

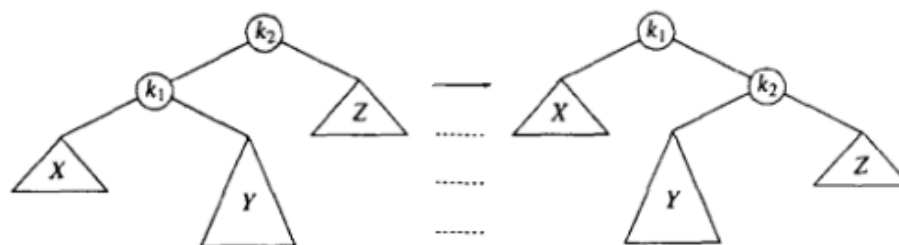


图 4-34 单旋转不能修复情形 2

如上图所见，Y的深度仍然在第三层没有动。

我们假设子树Y中有一项插入，**保证它是非空的**。因此，我们可以假设它**有一个根和两棵子树**。因此，我们将整棵树看作是四棵子树由三个节点连接

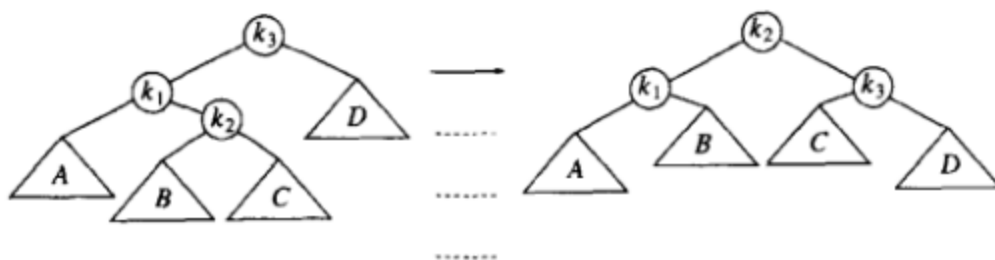


图 4-35 左-右双旋转修复情形 2

如图所示，恰好树B或树C中有一棵比D深两层（除非它们都为空），但是我们不能确定是哪一棵

为了重新平衡，我们不可以让 k_3 继续作为根，而 k_3 和 k_1 之间的旋转又无法解决问题，唯一的选择就是把 k_2 用作新的根。这迫使 k_1 是 k_2 的左儿子， k_3 是 k_2 的右儿子。

我们先左旋转 k_1 和 k_2 ，得到一个高度为3（若A或B不为NULL）的以 k_2 为根的子树
然后，再右旋转 k_2 和 k_3 得到一个平衡的AVL树

这棵树中若A和B为NULL，则单旋转 k_2 和 k_1 便可以解决。因为左旋转 k_2 后层数会减少
反之，若A或B其中一个存在，便要进行双旋转