

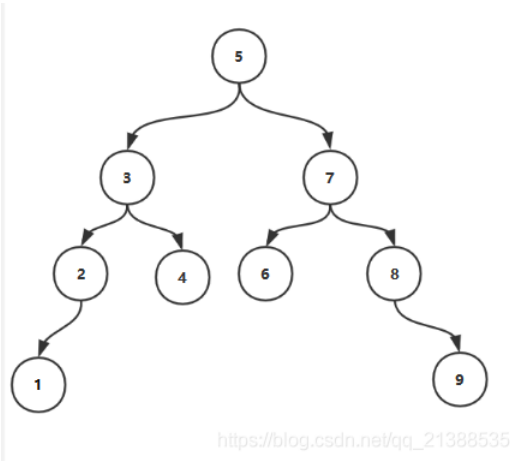
# AVL树删除，详细图解

AVL树删除节点的过程是，先找到该节点，然后进行删除。由于删除节点的位置不同，导致删除后节点进行移动的方式不同。删除节点的位置分为以下4类：

- 1.删除叶子结点。操作：直接删除，然后依次向上调整为AVL树。
- 2.删除非叶子节点，该节点只有左孩子。操作：该节点的值替换为左孩子节点的值，然后删除左孩子节点。【左孩子节点为叶子结点，所以删除左孩子节点的情况为第1种情况。】【为什么左孩子节点为叶子节点，因为删除节点前，该树是AVL树，由AVL树的定义知，每个节点的左右子树的高度差的绝对值<=1,由于该节点只有左孩子，没有右孩子，如果左孩子还有子节点，那么将不满足每个节点的左右子树的高度差的绝对值<=1，所以左孩子节点为叶子结点】
- 3.删除非叶子节点，该节点只有右孩子。操作：该节点的值替换为右孩子节点的值，然后删除右孩子节点。【右孩子节点为叶子结点，所以删除右孩子节点的情况为第1种情况。】【为什么右孩子节点为叶子节点？答案和第二种情况一样】
- 4.删除非叶子节点，该节点既有左孩子，又有右孩子。操作：该节点的值替换为该节点的前驱节点（或者后继节点），然后删除前驱节点（或者后继节点）。【前驱结点:在中序遍历中，一个节点的前驱结点，先找到该节点的左孩子节点，再找左孩子节点的最后一个右孩子节点。向左走一步，然后向右走到头。最后一个右孩子节点即为前驱节点】【后继节点：在中序遍历中，一个节点的后继结点，先找到该节点的右孩子节点，再找右孩子节点的最后一个左孩子节点。向右走一步，然后向左走到头。最后一个左孩子节点即为前驱节点】

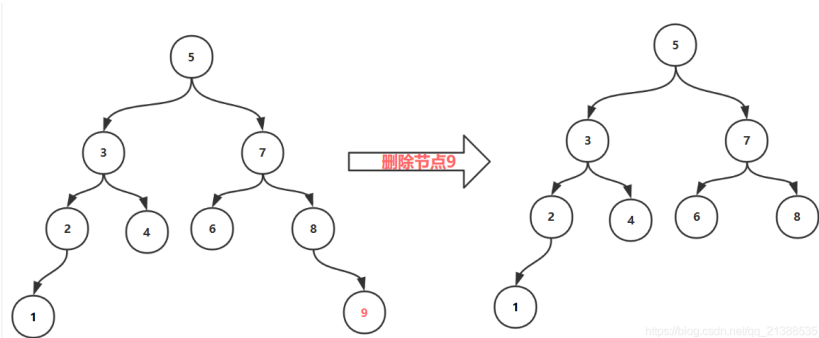
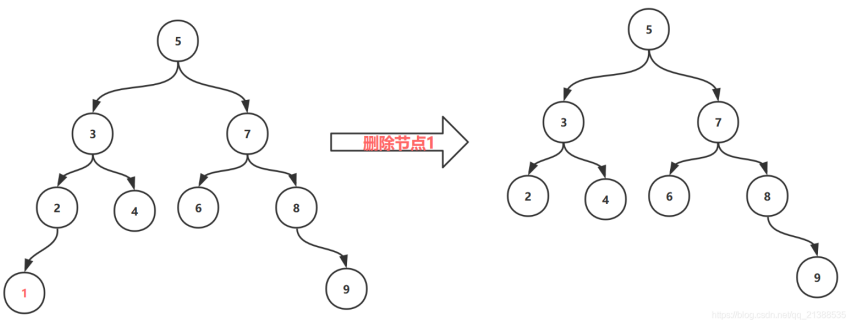
总结：对于非叶子节点的删除，最终都将转化为对叶子节点的删除。

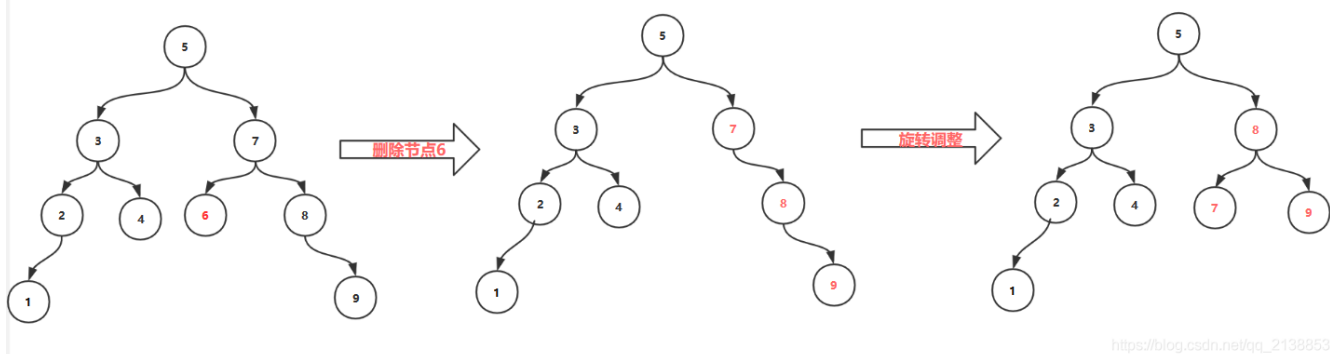
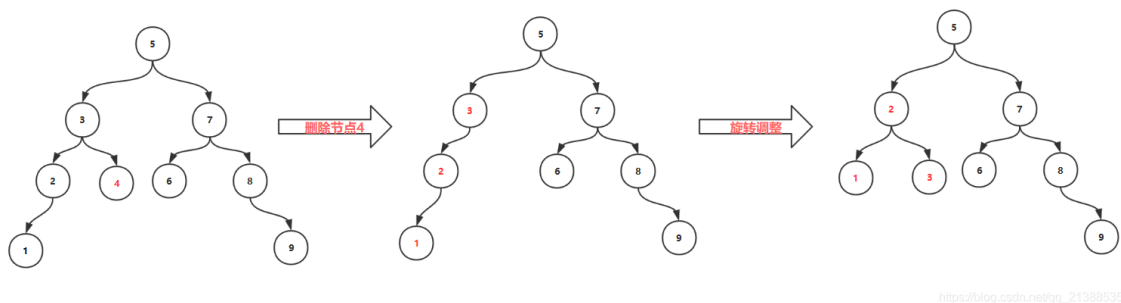
对下一棵AVL树，进行4种情况的删除。



## 第一种情况

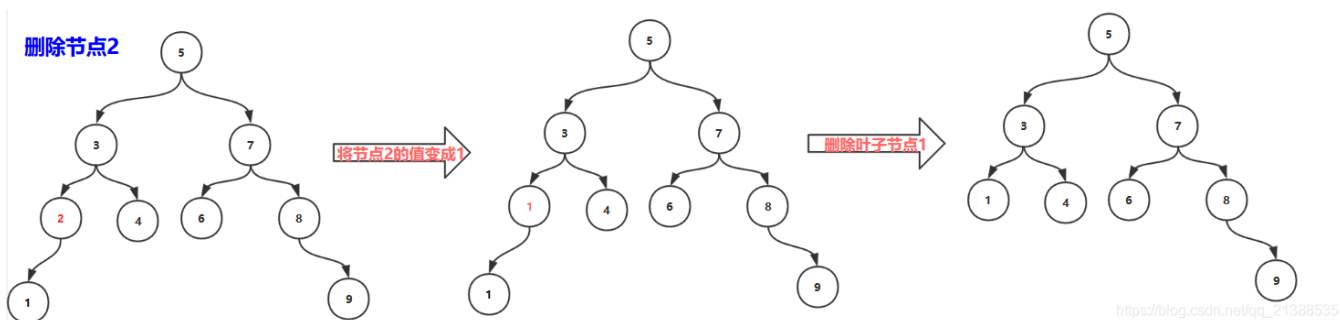
删除叶子节点1,节点9,节点4,节点6





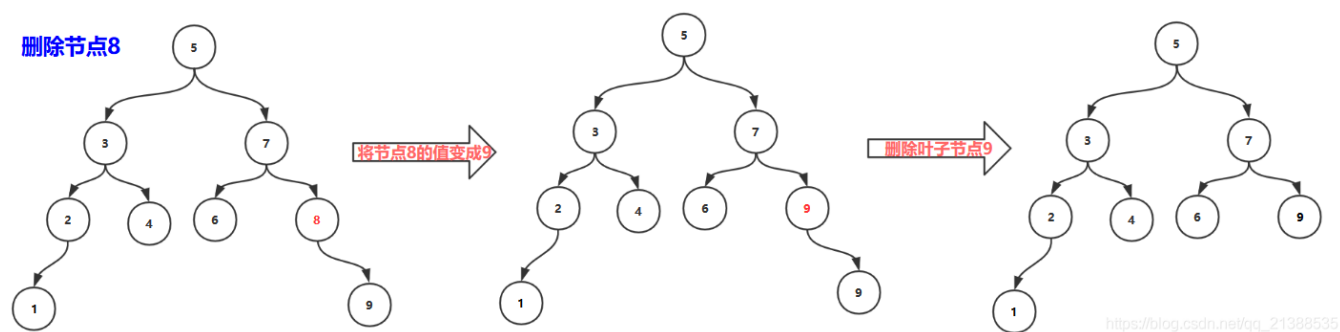
## 第二种情况：

删除非叶子节点，该节点只有左孩子



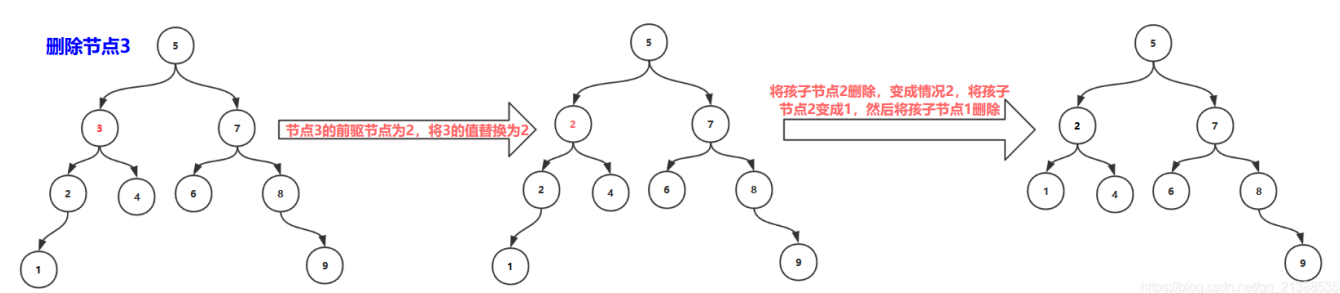
## 第三种情况

删除非叶子节点，该节点只有右孩子

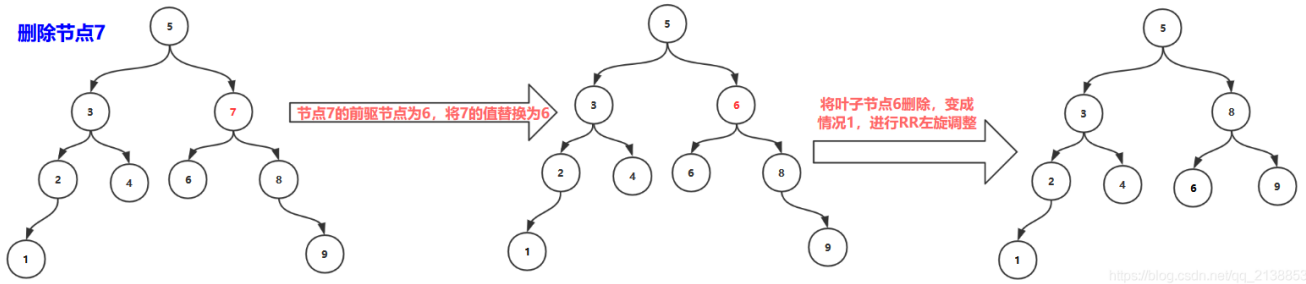


## 第四种情况

删除非叶子节点（节点3，节点7，节点5），非叶子节点既有左孩子，又有右孩子

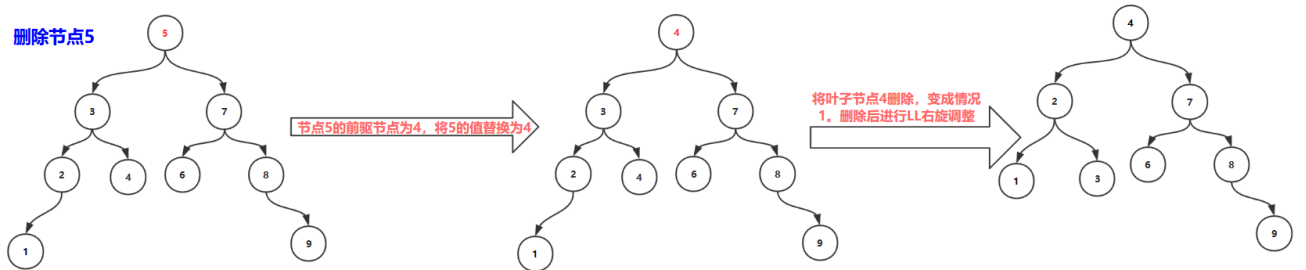


### 删除节点7



[https://blog.csdn.net/q\\_21388535](https://blog.csdn.net/q_21388535)

### 删除节点5



[https://blog.csdn.net/q\\_21388535](https://blog.csdn.net/q_21388535)