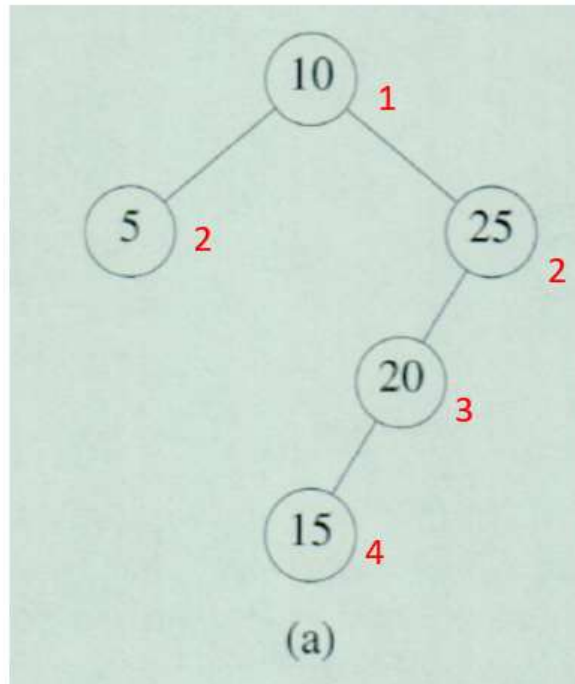


Chapter 09 搜尋

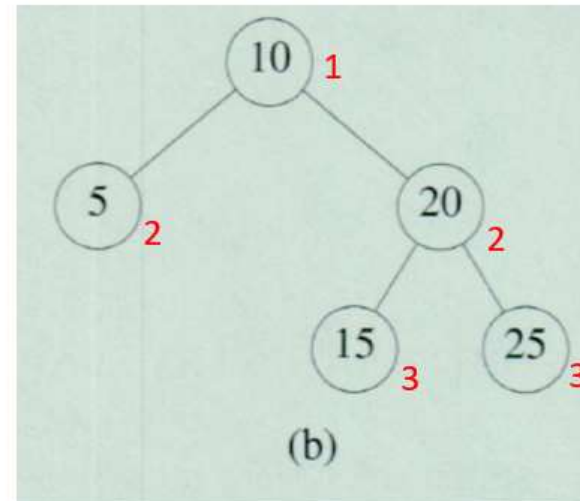
(Searching)(2)

二元搜尋樹

- 二元搜尋樹的搜尋
 - 遞迴版（請參考5.2的投影片）
 - 非遞迴版（請參考5.2的投影片）



平均比較 $(1+2+2+3+4)/5 = 2.4$ 次

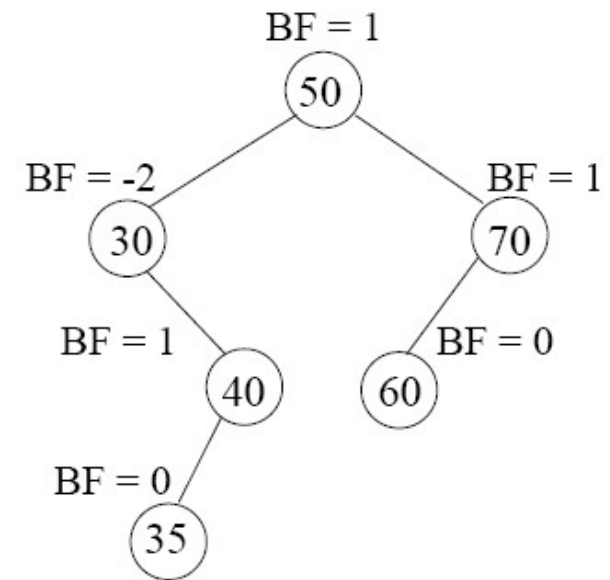


平均比較 $(1+2+2+3+3)/5 = 2.2$ 次

- 二元搜尋樹為完整二元樹，則平均及最大搜尋時間會最小

高度平衡二元搜尋樹 (Height Balanced Binary Tree, AVL Tree)

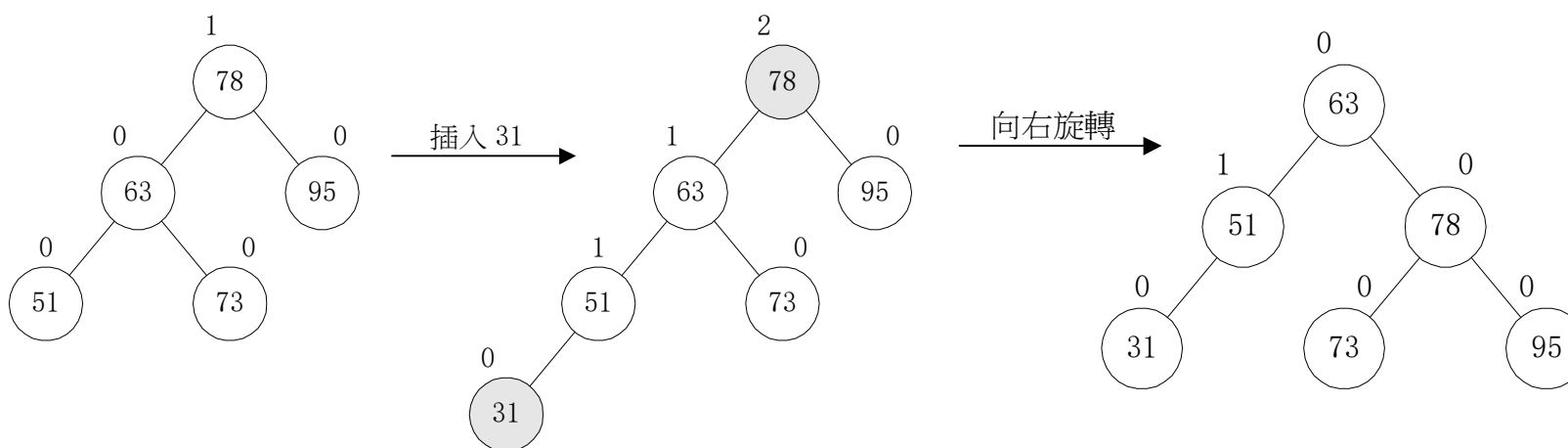
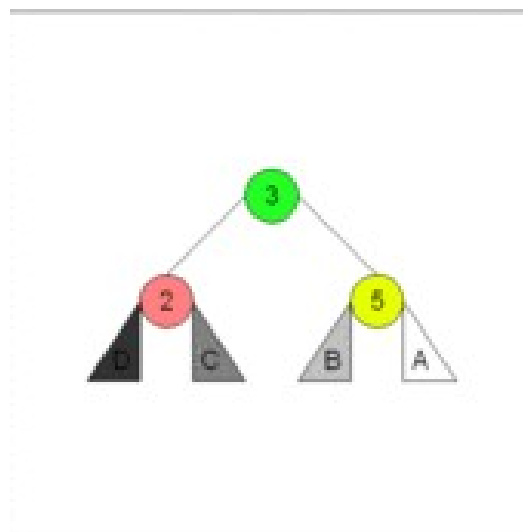
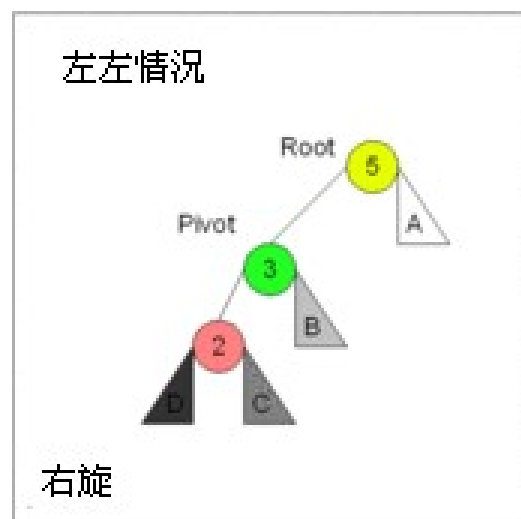
- 空樹(empty tree)是高度平衡二元樹。
- 假使T不是空的二元樹， T_L 和 T_R 分別是此二元樹的左子樹和右子樹，若符合下列二個條件，則稱T為高度平衡二元樹，也稱為**AVL-Tree**。
 1. T_L 和 T_R 亦是高度平衡二元樹，
 2. $|h_L - h_R| \leq 1$ ，其中 h_L 及 h_R 分別是 T_L 和 T_R 的高度；
 3. $h_L - h_R$ 為平衡因子(balanced factor, BF)在AVL-Tree中，每一節點的平衡因子為-1、0或1。



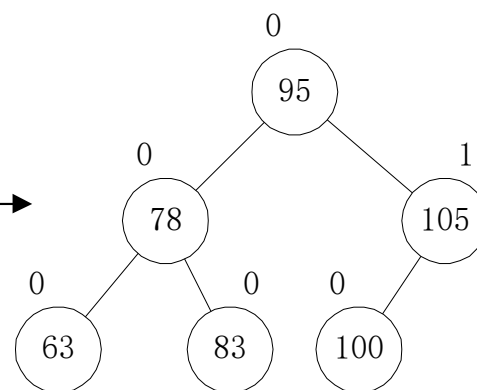
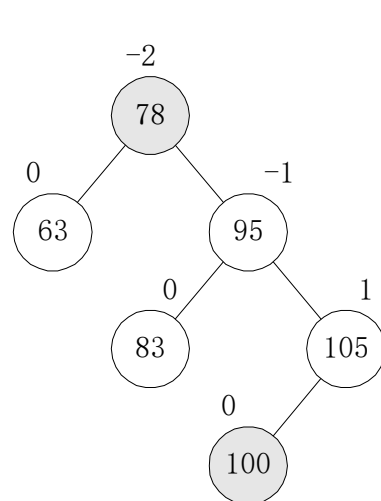
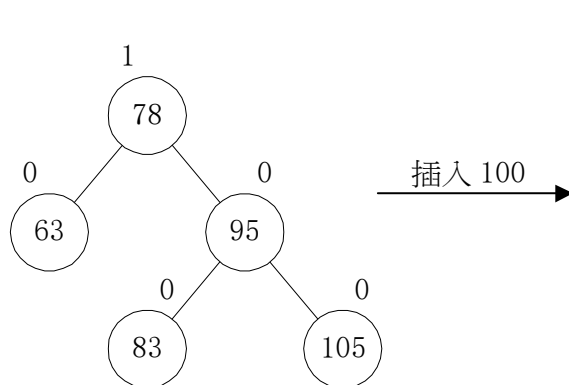
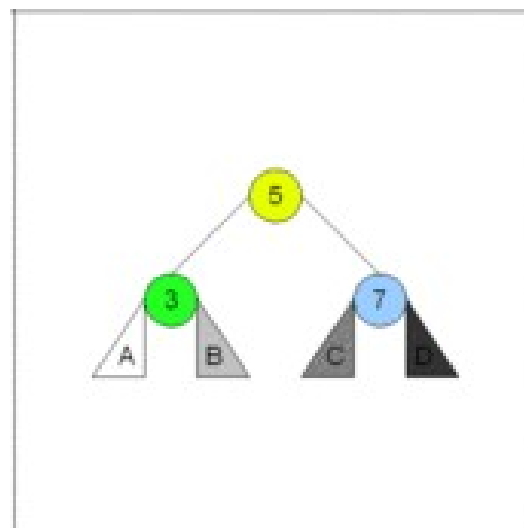
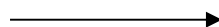
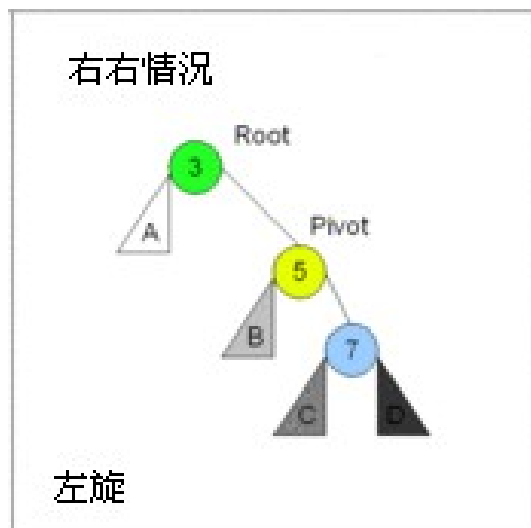
AVL Tree的加入

- 高度平衡二元搜尋樹在加入或刪除後，可能會造成不平衡，此時可利用LL，RR，LR，RL等四種不同的調整方式，使其符合AVL-tree的定義。
- LL：加入的新節點N在節點p的左邊的左邊。
- RR：加入的新節點N在節點p的右邊的右邊。
- LR：加入的新節點N在節點p的左邊的右邊。
- RL：加入的新節點N在節點p的右邊的左邊。

AVL Tree的加入 - LL

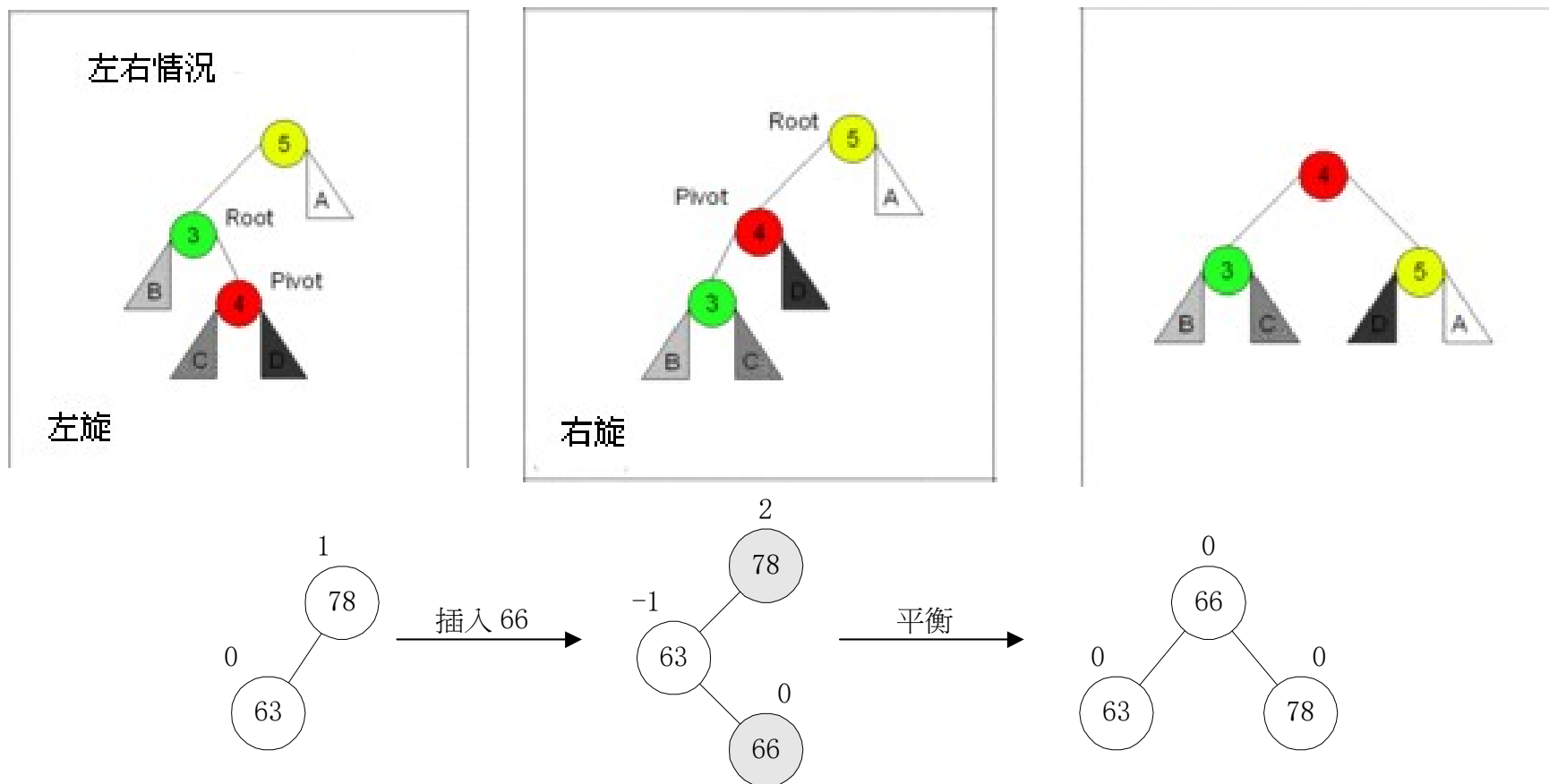


AVL Tree的加入 - RR

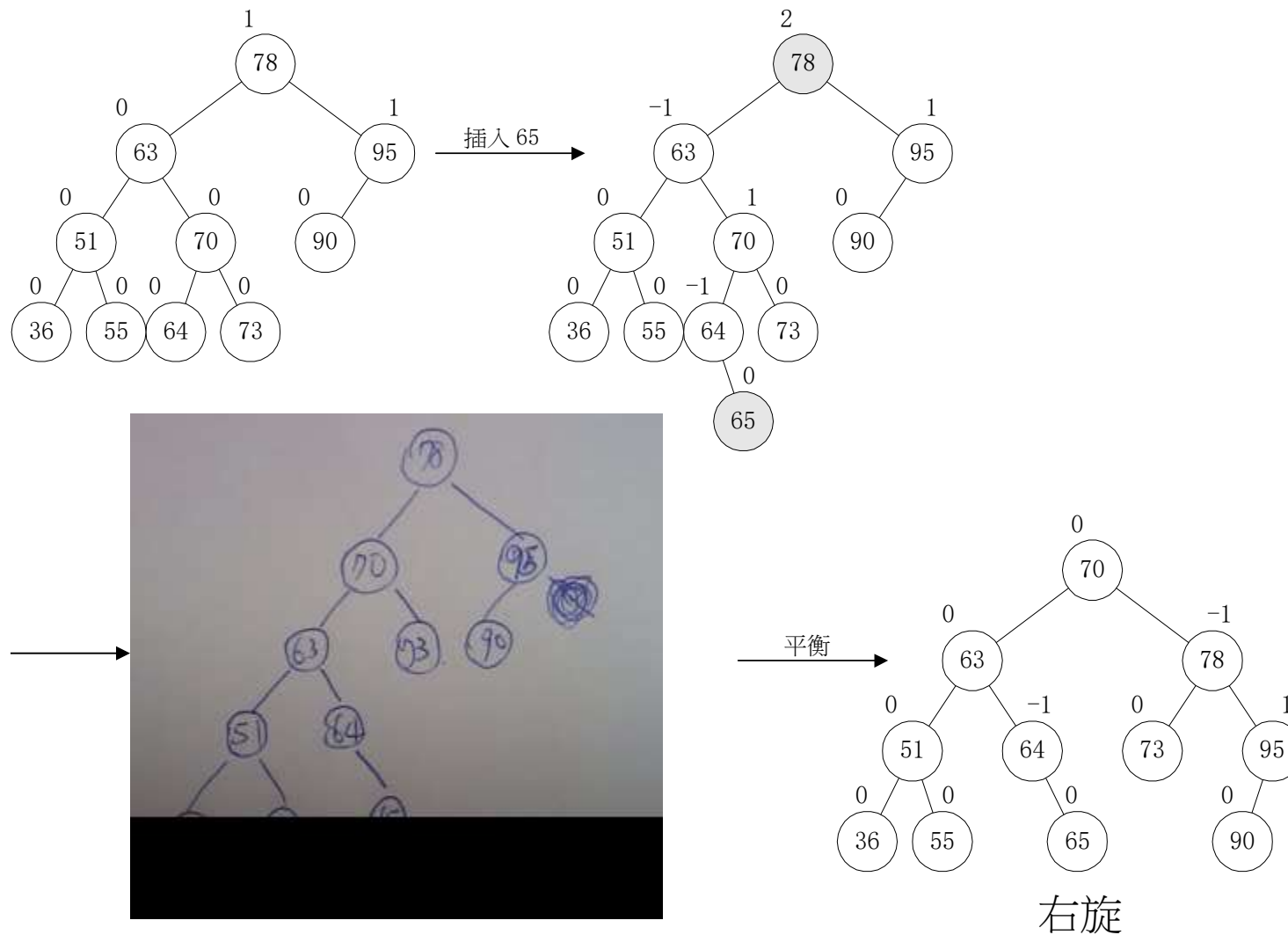


AVL Tree的加入 - LR (1)

- LR 型 AVL 樹



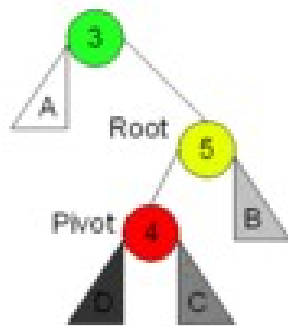
AVL Tree的加入 - LR (2)



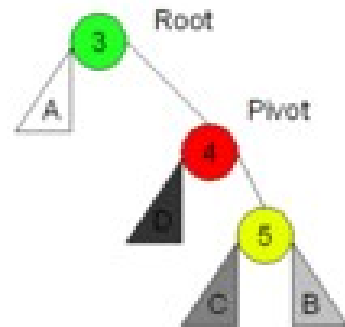
AVL Tree的加入 - RL (1)

• RL 型 AVL 樹

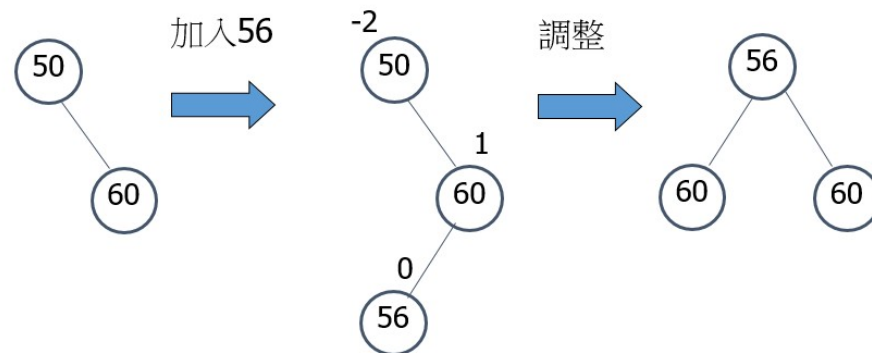
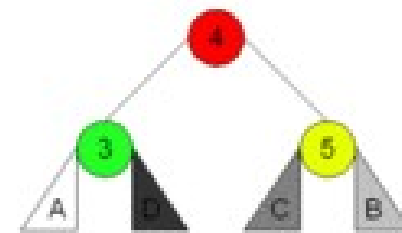
右左情況



右旋



左旋



AVL Tree的加入 - RL (2)

