9.2.1 二叉排序树和平衡二叉树

- 何谓“平衡二叉树”？
- 如何构造“平衡二叉树”
- 平衡二叉树的查找性能分析

我们希望所建的二叉排序树均为平衡二叉树

以下所研究的平衡二叉树均为平衡二叉排序树

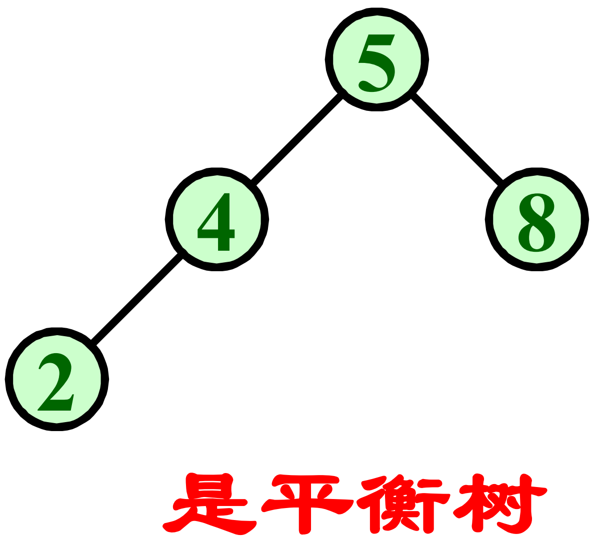
平衡二叉树(AVL树)

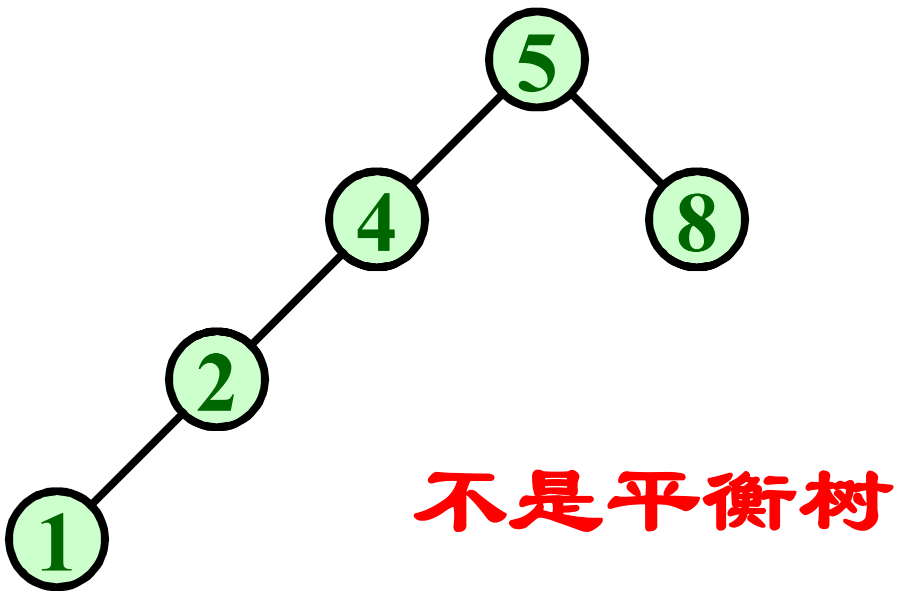
- 空树，或者是具有下列性质的二叉树：

  (1) 左右子树都是平衡二叉树

  (2) 左右子树高度之差的绝对值不超过1

- 树中每个结点的左、右子树高度之差的绝对值不大于1。

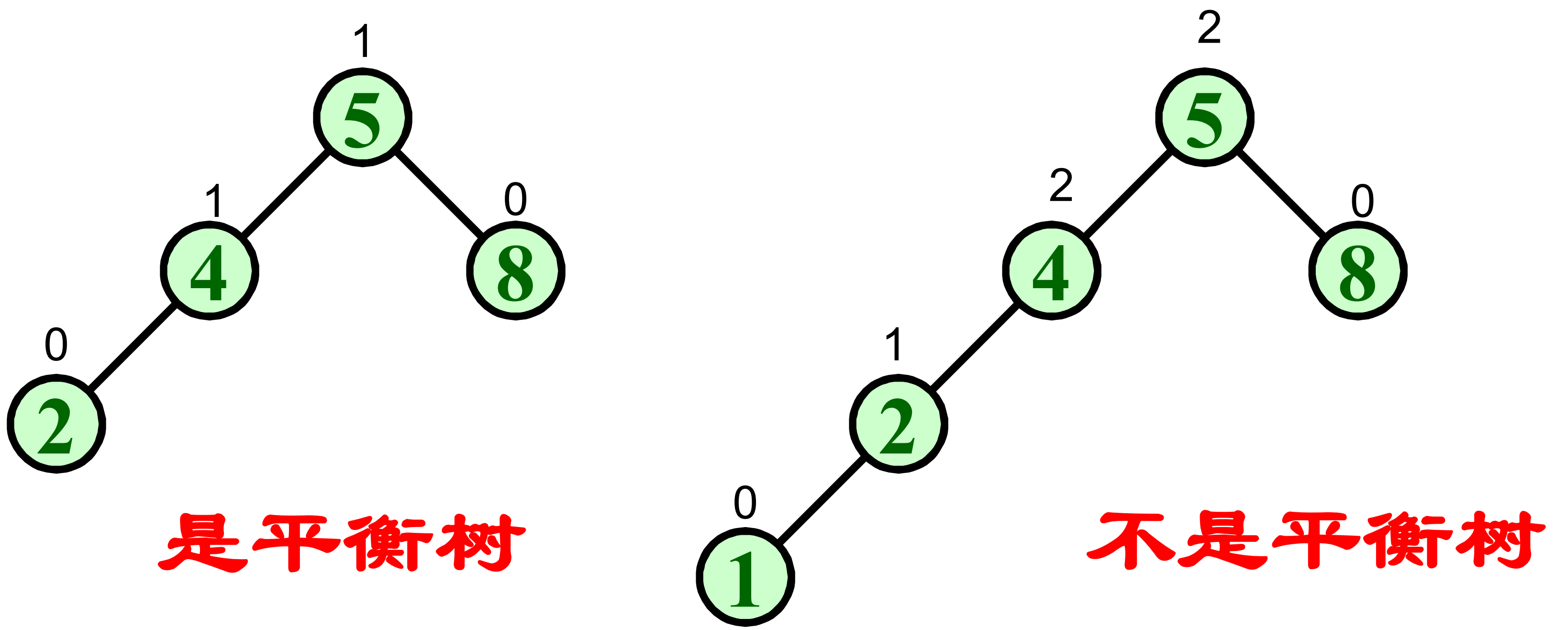




平衡二叉树(AVL树)

•结点的平衡因子BF=结点的左子树深度-右子树深度

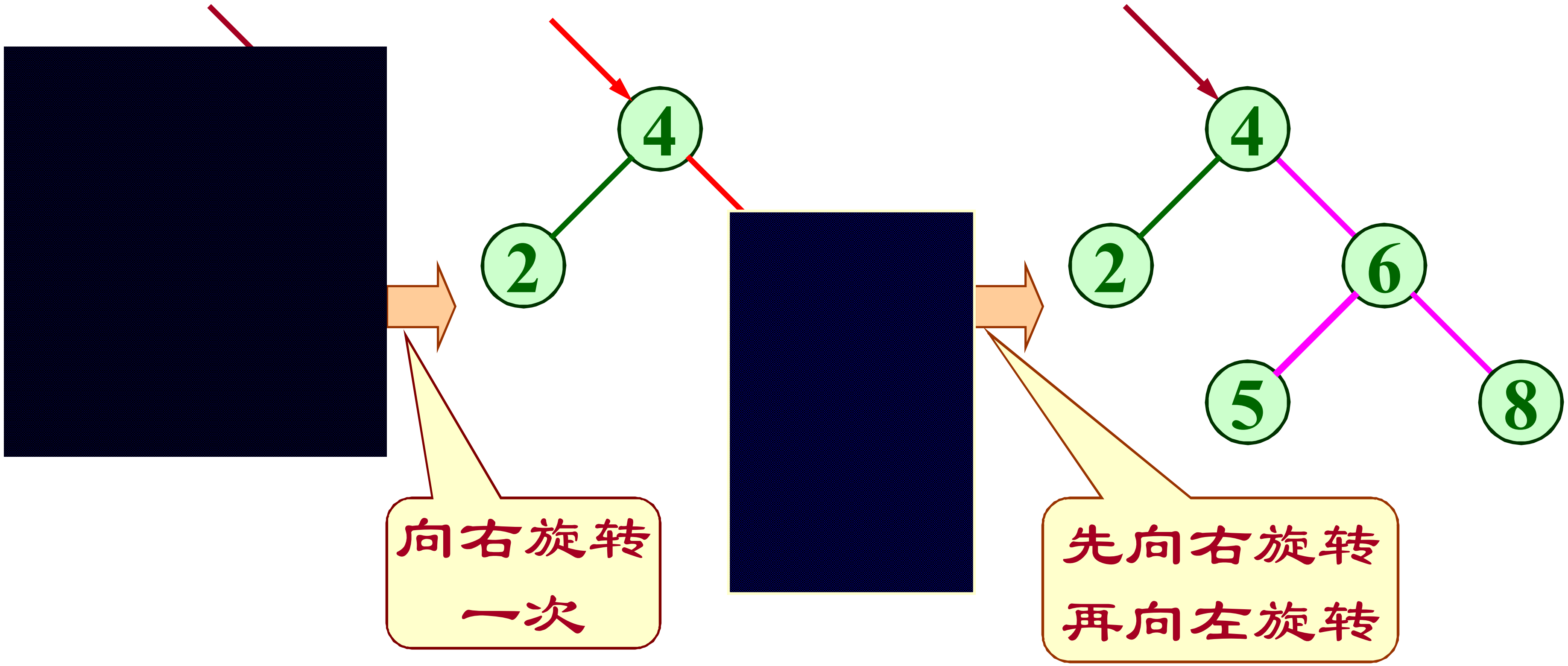
- 平衡二叉树每个结点的平衡因子的绝对值不超过1



构造平衡二叉（查找）树的方法是：

在插入过程中，采用平衡旋转技术。

例如:依次插入的关键字为5, 4, 2, 8, 6, 9



平衡旋转技术

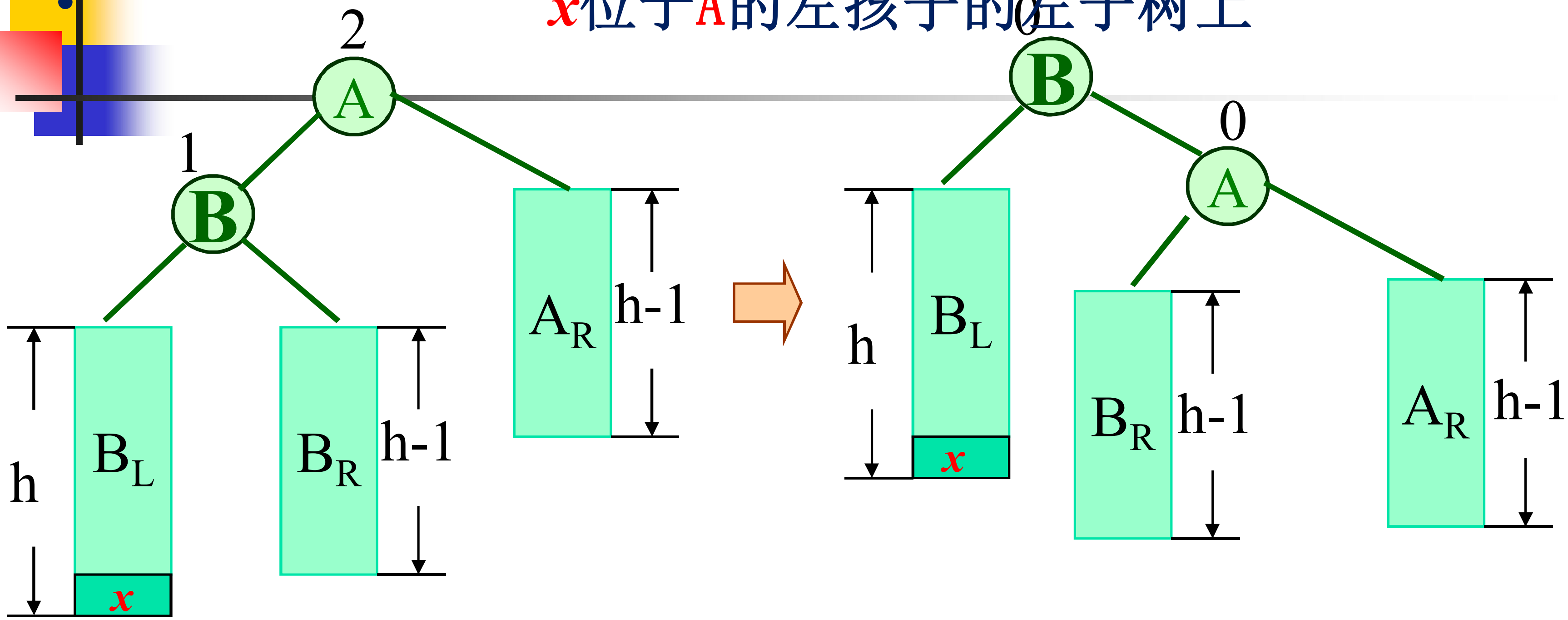
如果在一棵AVL树中插入一个新结点，就有可能造成失衡，此时必须**重新调整树的结构**，使之恢复平衡。我们称调整平衡过程为**平衡旋转**。

平衡旋转可以归纳为四类：

- LL平衡旋转--单向右旋
- RR平衡旋转--单向左旋
- LR平衡旋转--先左旋后右旋
- RL平衡旋转--先右旋后左旋

•LL型—单向右旋：A为离新插结点*x*最近的失去平衡的结点

*x*位于A的左孩子的左子树上



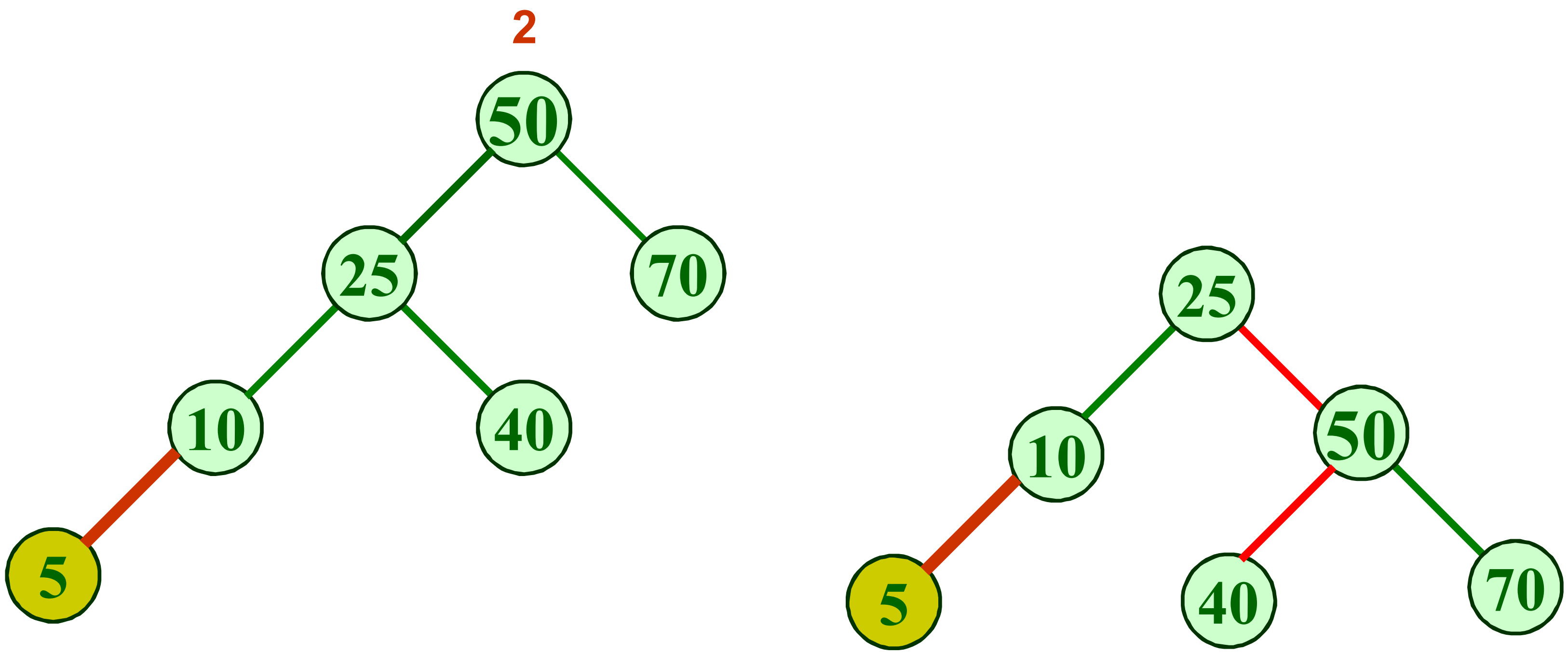
在单向右旋平衡处理后BF(B)由1变为0，BF(A)由2变为0

由于该子树高度不变，不会影响其祖先和其他结点的平衡因子！！！

要知道：在一平衡二叉树中插入一结点，只会影响其祖先结点的平衡因子！

LL型—单向右旋

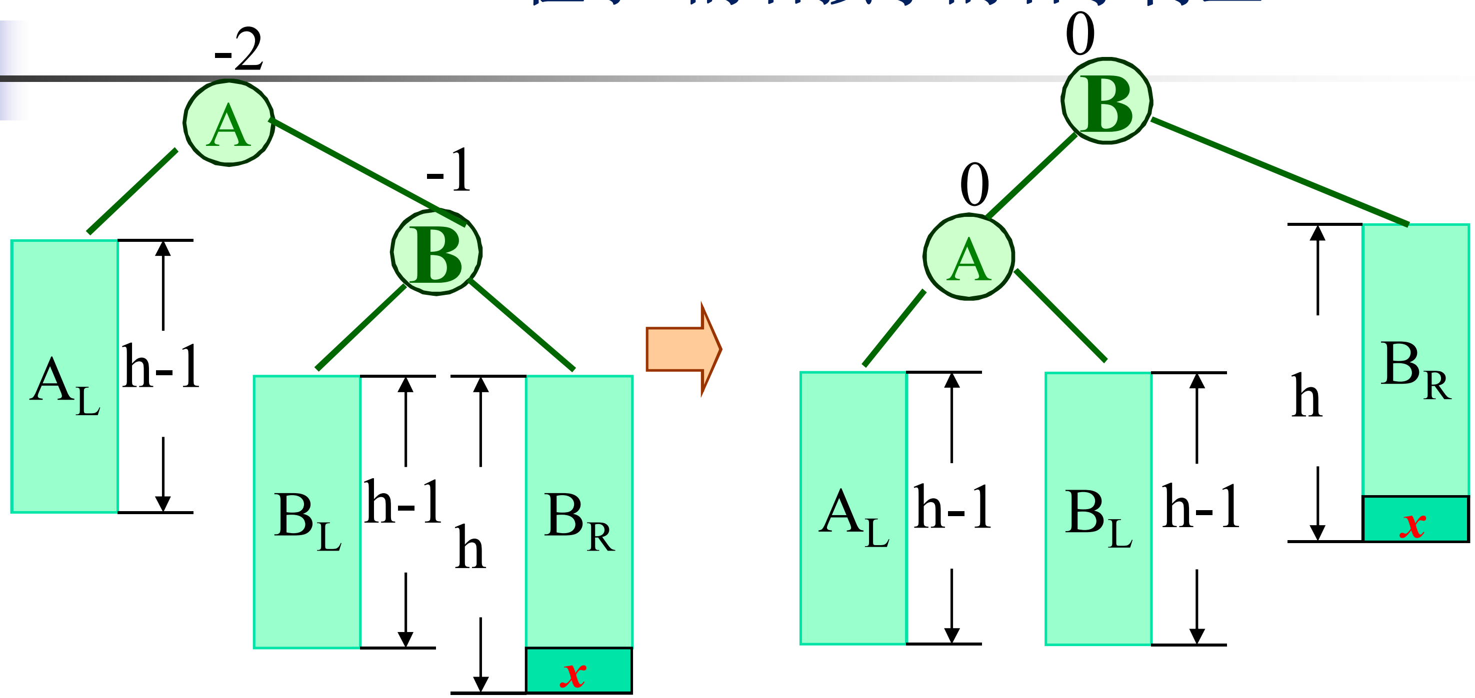
离新插结点5最近的失去平衡的是结点50！



在一平衡二叉排序树中插入5，插入后保证还是平衡二叉排序树

• RR型——单向左旋：A为离新插结点*x*最近的失去平衡的结点

• *x*位于A的右孩子的右子树上

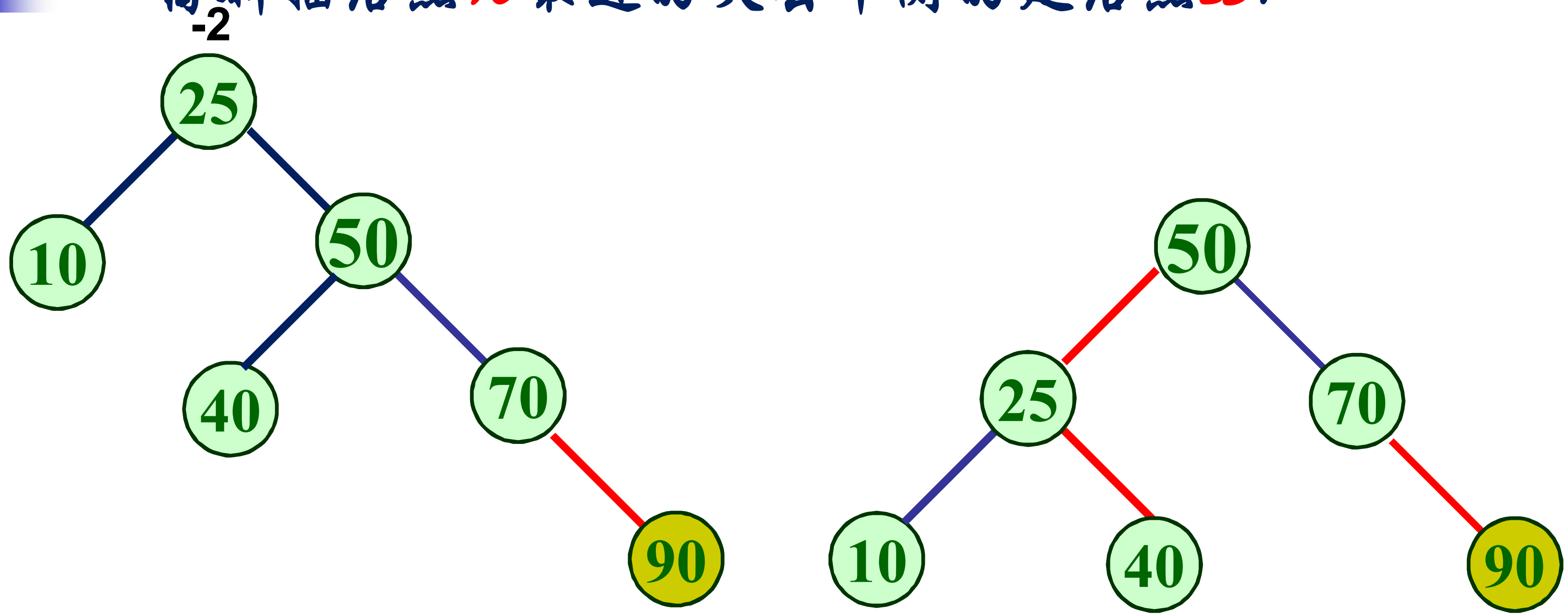


在单向左旋平衡处理后BF(B)由-1变为0，BF(A)由-2变为0

由于该子树高度不变，不会影响其祖先和其他结点的平衡因子！！！

RR型——单向左旋

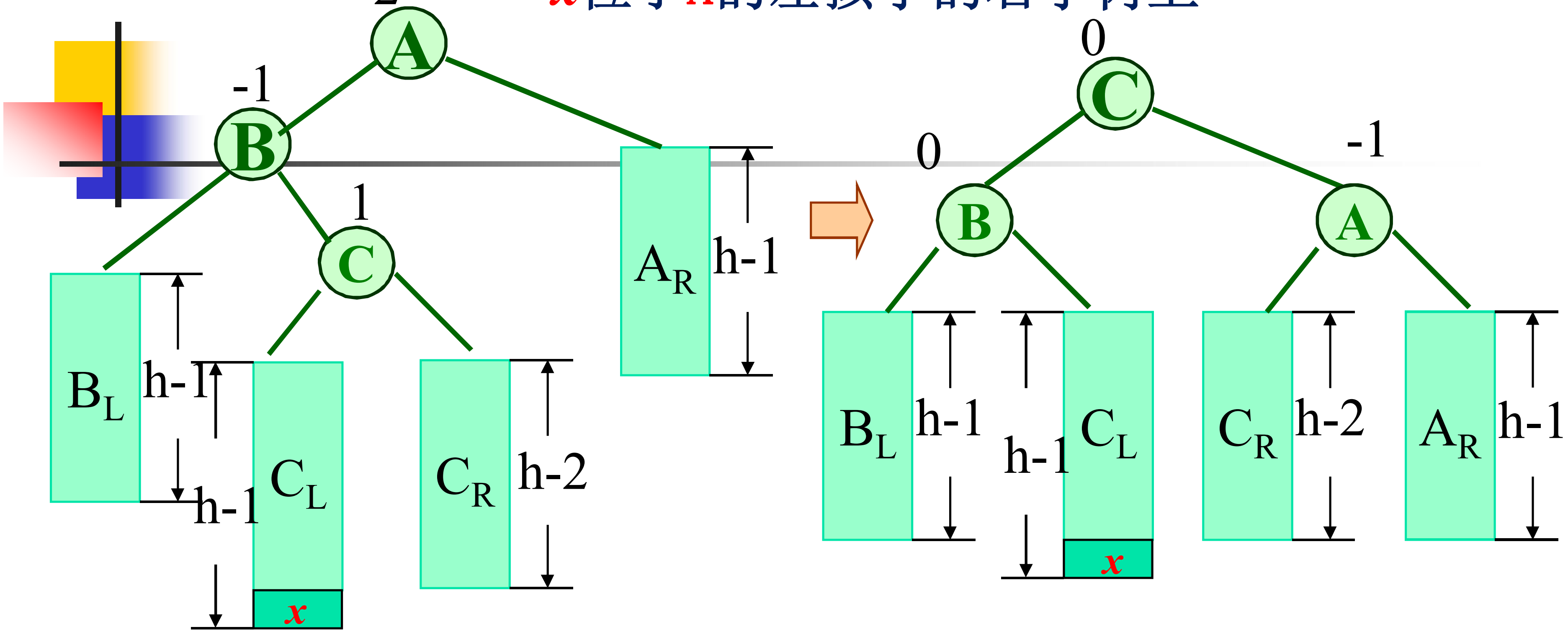
离新插结点90最近的失去平衡的是结点25!



在一平衡二叉排序树中插入90，插入后保证还是平衡二叉排序树

•LR型—先左旋后右旋：A为离新插结点*x*最近的失去平衡的结点

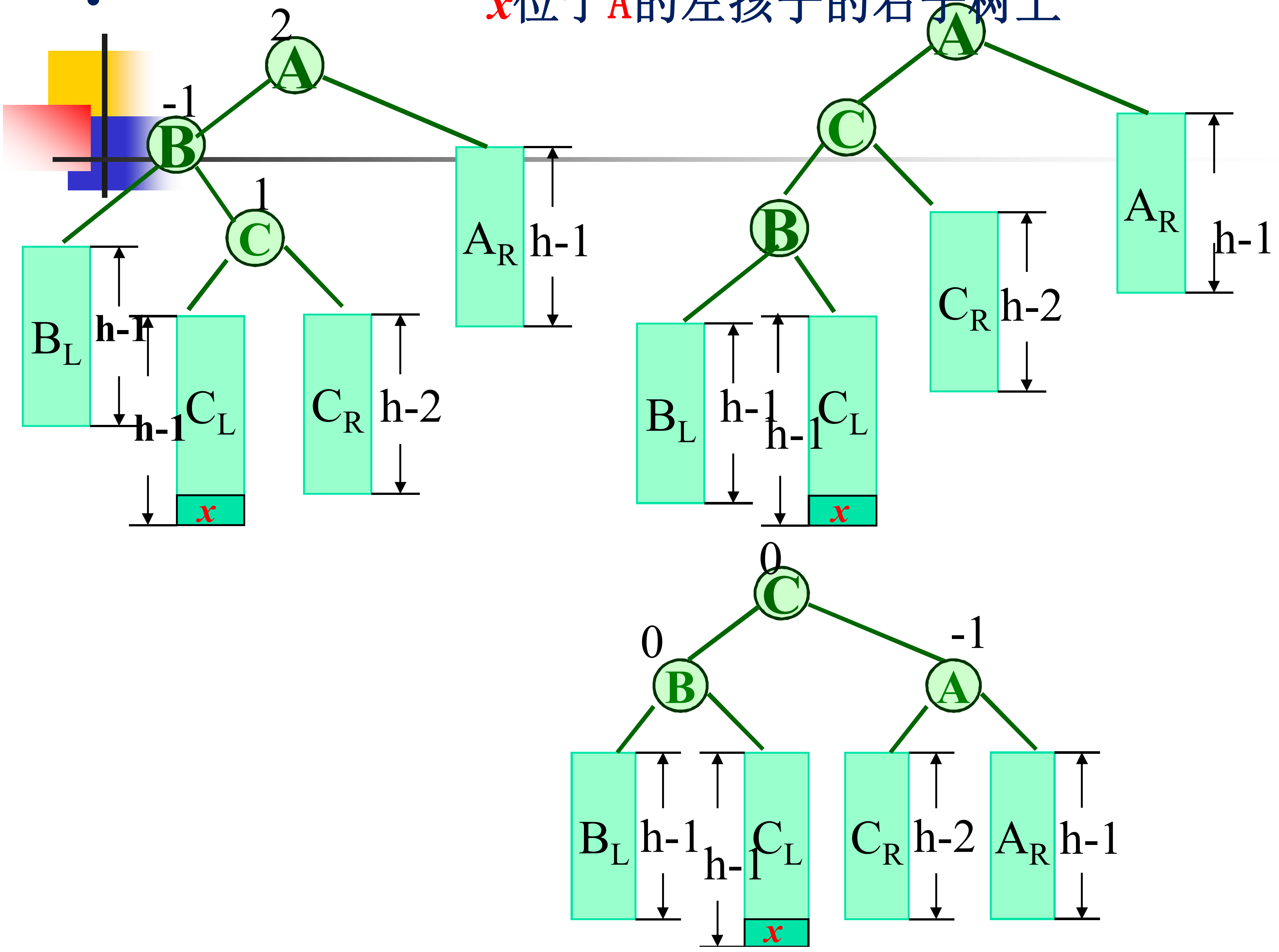
• *x*位于A的左孩子的右子树上



在双向旋转平衡处理后BF(A)由2变为-1，BF(B)由-1变为0

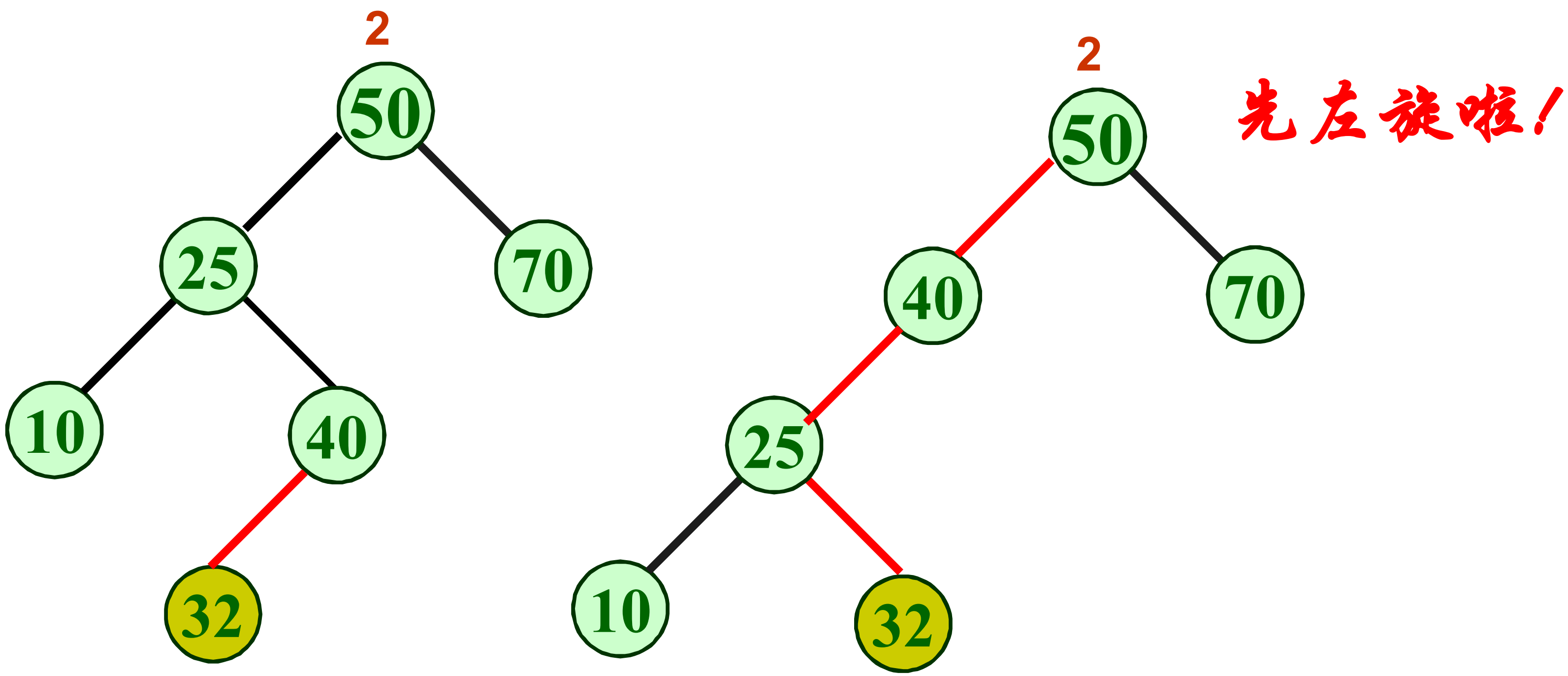
BF(C)由1变为0

- LR型—先左旋后右旋：A为离新插结点*x*最近的失去平衡的结点
- *x*位于A的左孩子的右子树上



LR型—先左旋后右旋

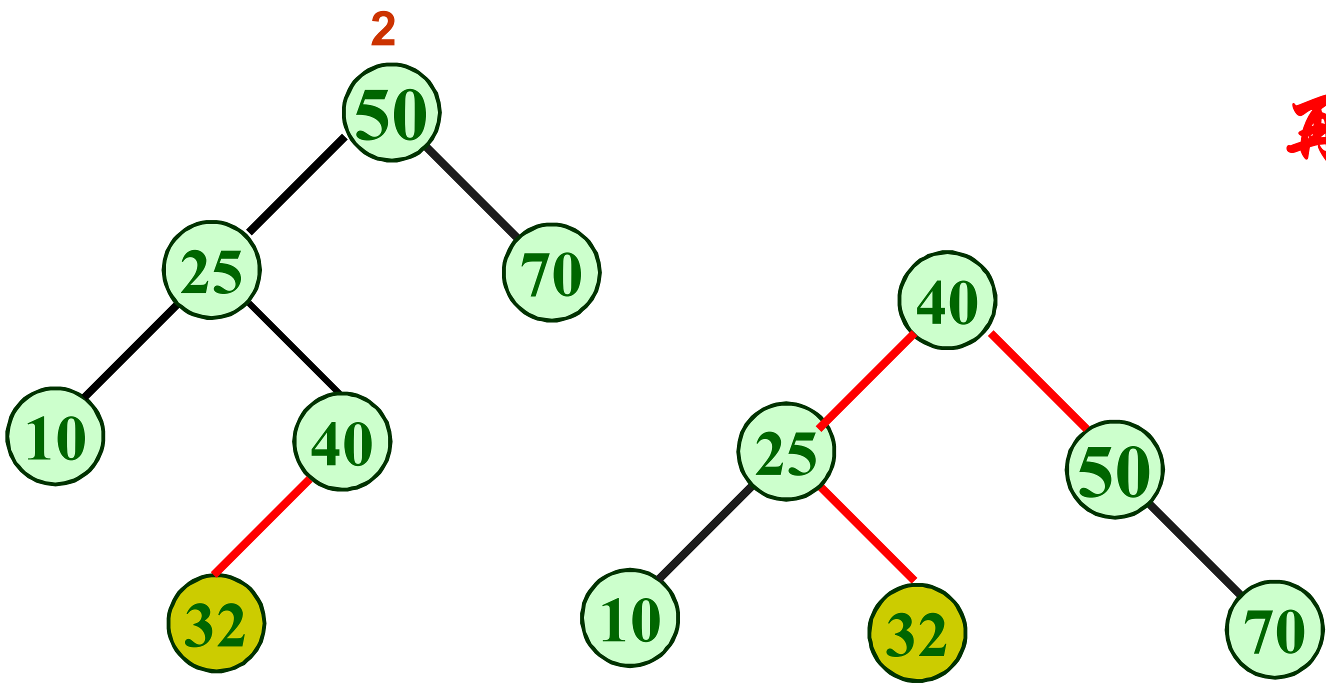
离新插结点32最近的失去平衡的是结点50！



在一平衡二叉排序树中插入32，插入后保证还是平衡二叉排序树

LR型—先左旋后右旋

离新插结点32最近的失去平衡的是结点50!

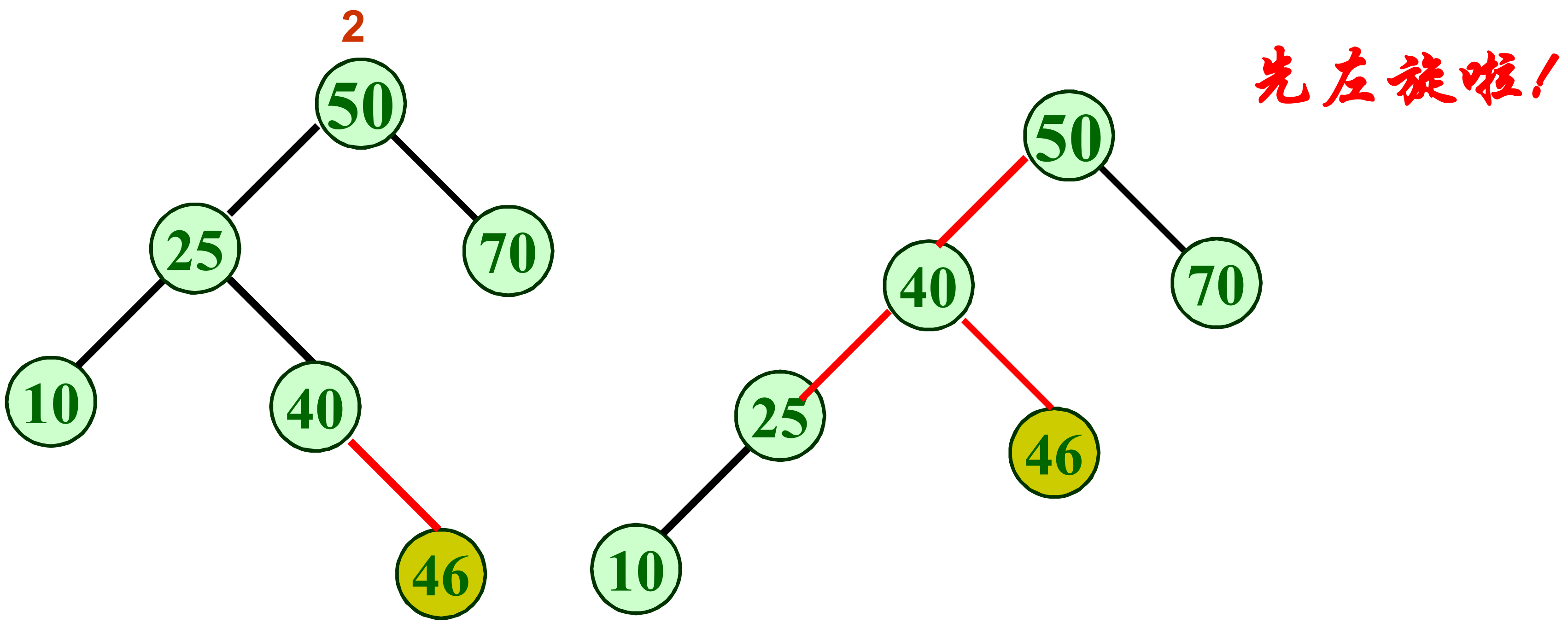


再右旋啦!

在一平衡二叉排序树中插入32，插入后保证还是平衡二叉排序树

LR型—先左旋后右旋

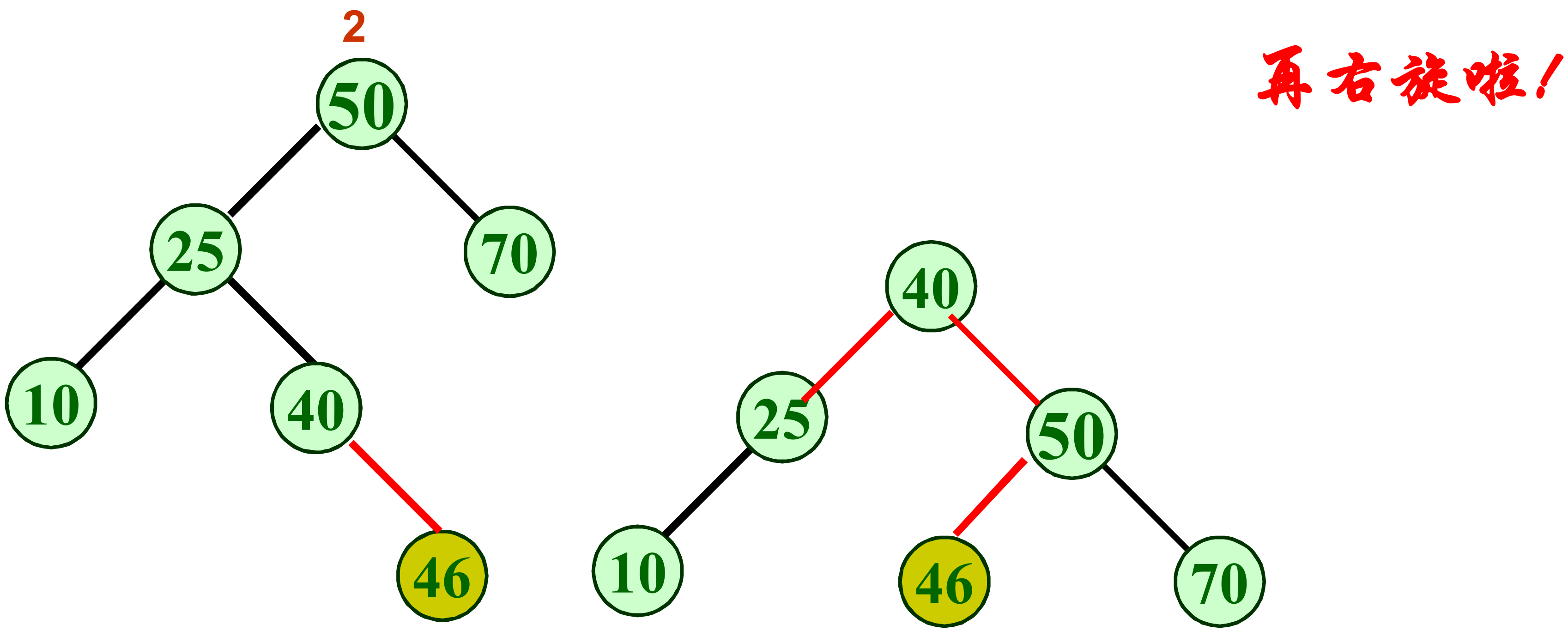
离新插结点46最近的失去平衡的是结点50!



在一平衡二叉排序树中插入46，插入后保证还是平衡二叉排序树

LR型—先左旋后右旋

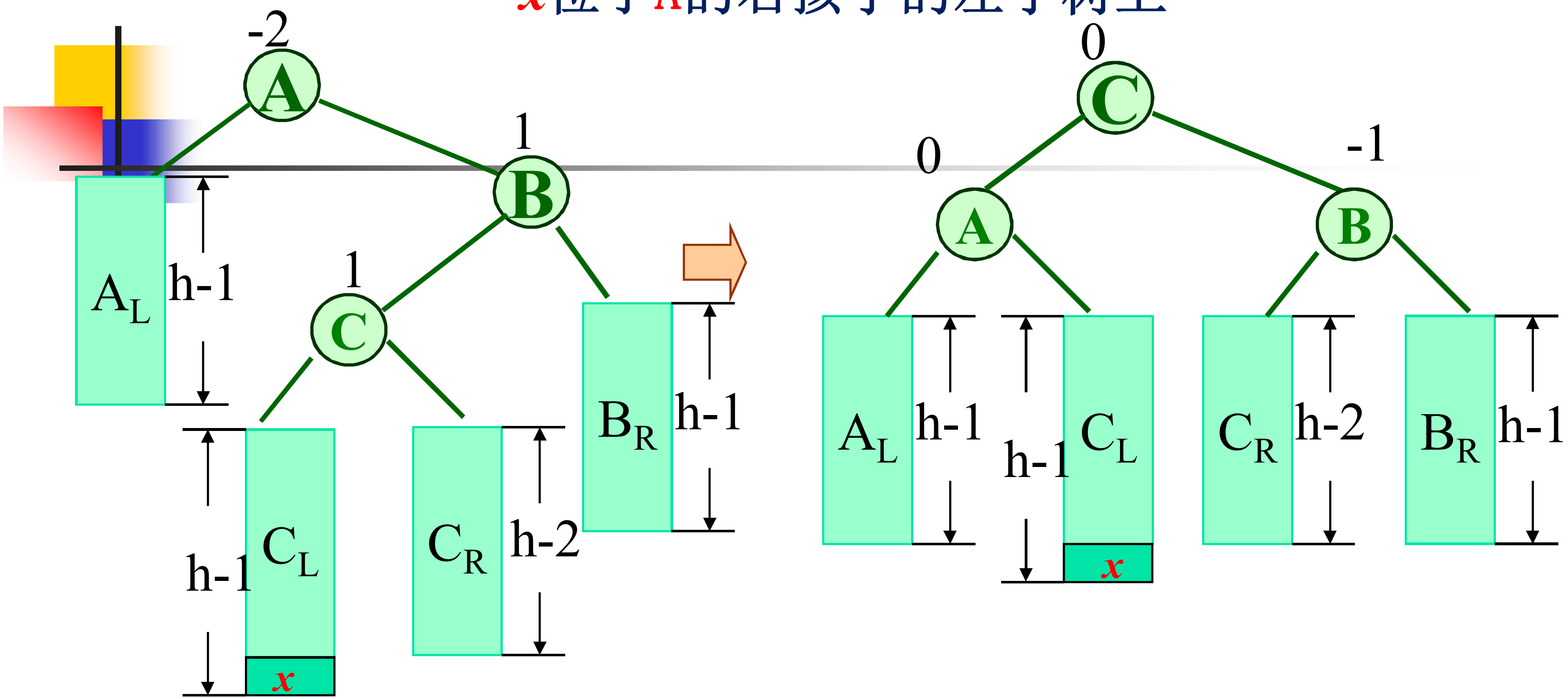
离新插结点46最近的失去平衡的是结点50!



在一平衡二叉排序树中插入46，插入后保证还是平衡二叉排序树

•RL型—先右旋后左旋：A为离新插结点*x*最近的失去平衡的结点

• *x*位于A的右孩子的左子树上

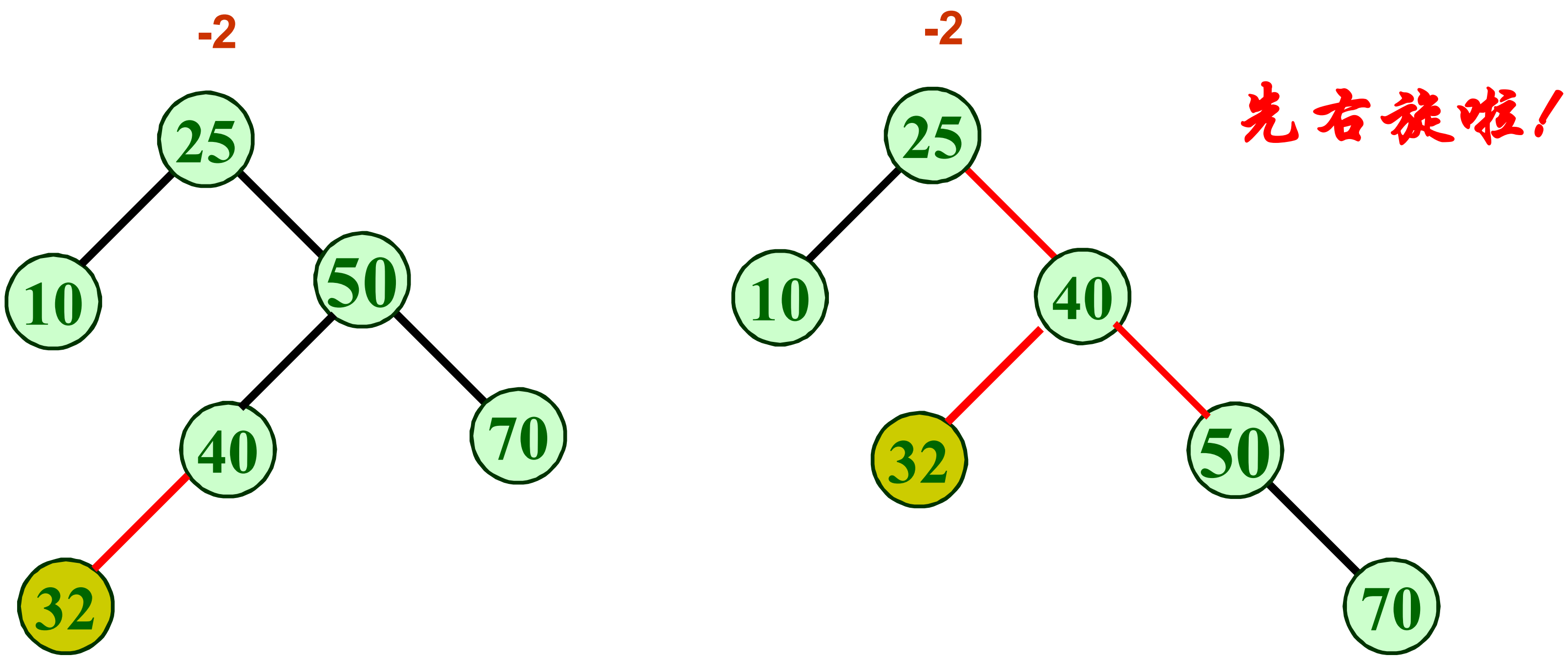


在双向旋转平衡处理后BF(A)由-2变为0，BF(B)由1变为-1

BF(C)由1变为0

RL型—先右旋后左旋

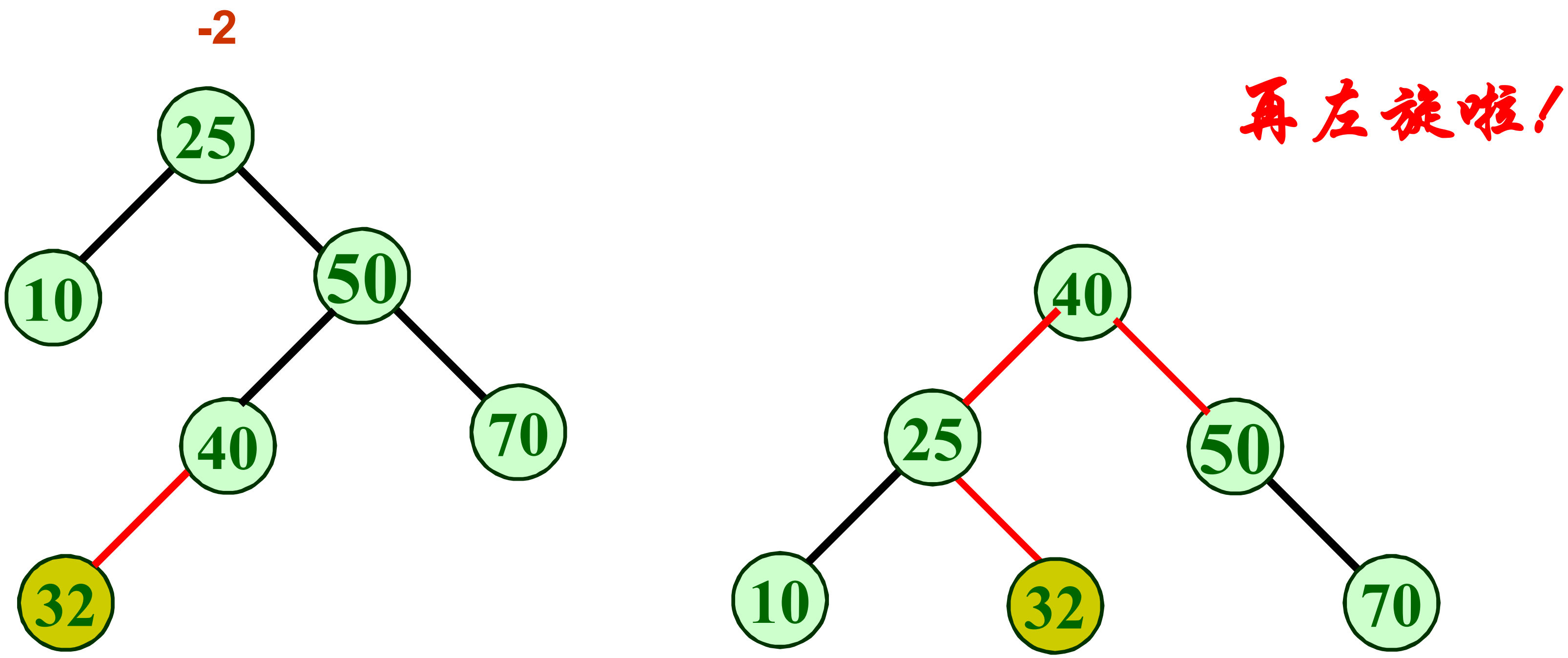
离新插结点32最近的失去平衡的是结点25!



在一平衡二叉排序树中插入32，插入后保证还是平衡二叉排序树

RL型—先右旋后左旋

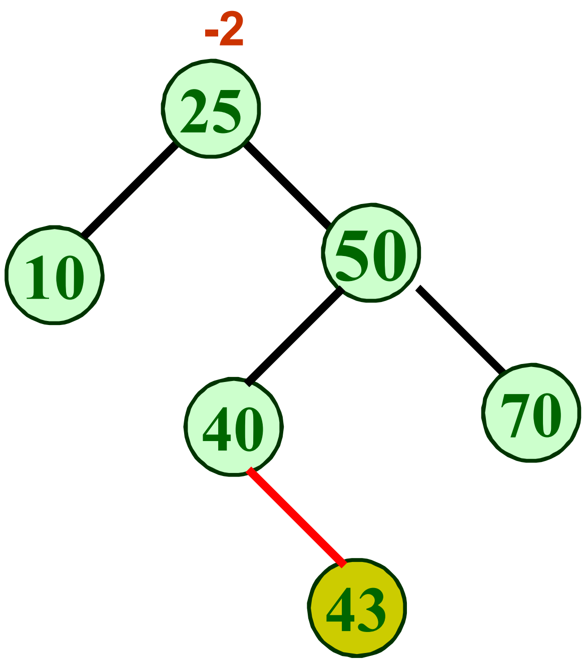
离新插结点32最近的失去平衡的是结点25!

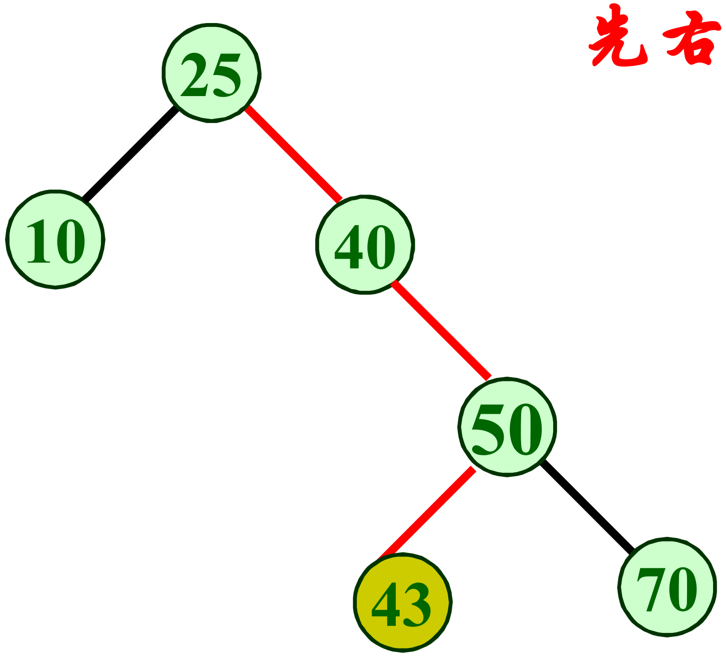


在一平衡二叉排序树中插入32，插入后保证还是平衡二叉排序树

RL型—先右旋后左旋

离新插结点43最近的失去平衡的是结点25!



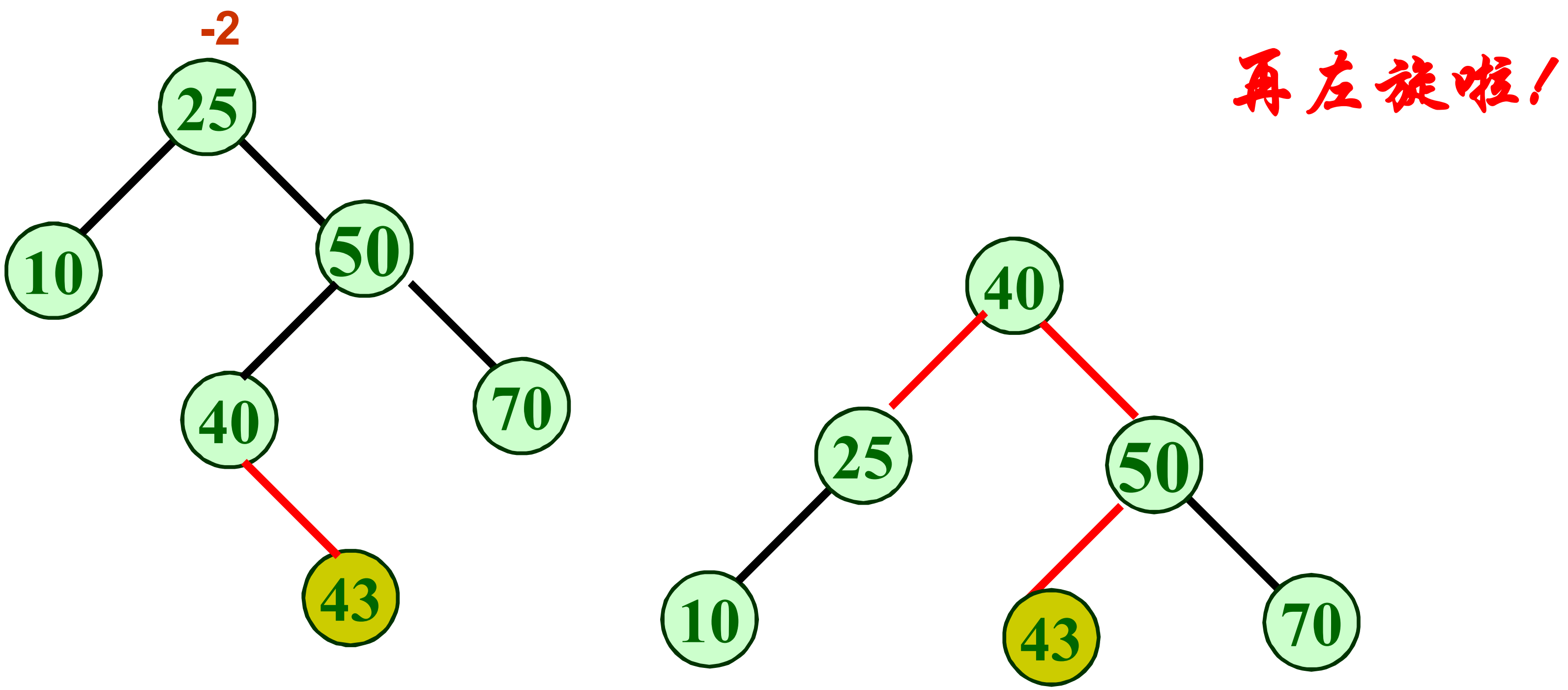


先右旋啦!

在一平衡二叉排序树中插入43，插入后保证还是平衡二叉排序树

RL型—先右旋后左旋

离新插结点43最近的失去平衡的是结点25!



在一平衡二叉排序树中插入43，插入后保证还是平衡二叉排序树

旋转操作特点

1. 对不平衡的最小子树操作
2. 旋转后子树根节点平衡因子为0
3. 旋转后子树深度不变故不影响全树，也不影响插入路径上所有祖先结点的平衡度

平衡树查找的性能

- 查找的时间复杂度为O(log$n$)

平衡树的查找性能分析

在平衡树上进行查找的过程和二叉排序树相同，因此，查找过程中和给定值进行比较的关键字的个数不超过平衡树的深度。

问：含 n 个关键字的二叉平衡树可能达到的**最大深度**是多少?

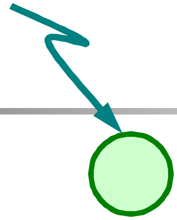
先看几个具体情况:

$n = 0$

空树

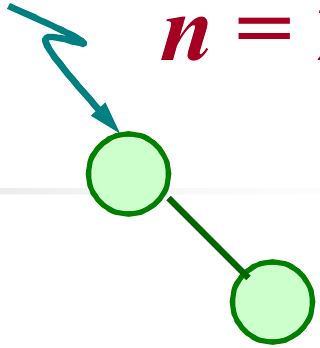
最大深度为 0

$n = 1$



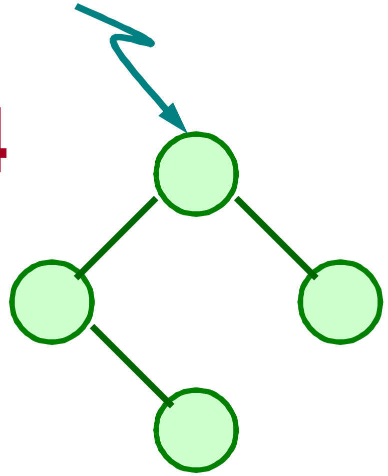
最大深度为 1

$n = 2$



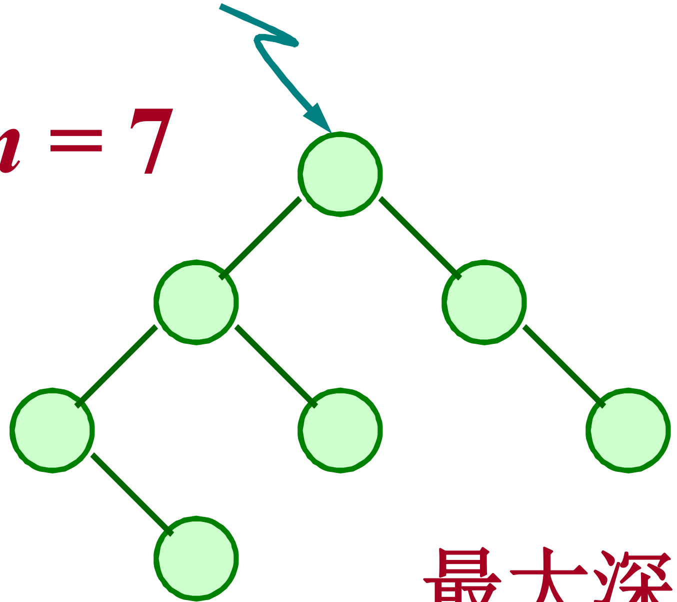
最大深度为 2

$n = 4$



最大深度为 3

$n = 7$



最大深度为 4

反过来问，深度为 h 的二叉平衡树中所含结点的最小值 N_h 是多少?

$h=0 \quad N_0=0 \qquad h=1 \quad N_1=1$

$h=2 \quad N_2=2 \qquad h=3 \quad N_3=4$

一般情况下 $\quad N_h = N_{h-1} + N_{h-2} + 1$

利用归纳法可证得 $\quad N_h = F_{h+2} - 1$

$$\varphi = \frac{1+\sqrt{5}}{2}$$

- $F_0=1, F_1=1, F_{n+1}=F_{n-1}+F_n$

由此推得，深度为 *h* 的二叉平衡树中所含结点的最小值 $N_h = \varphi^{h+2}/5 - 1$。

反之，含有 *n* 个结点的二叉平衡树能达到的最大深度 $h_n = \log_\varphi(\sqrt{5}\,(n+1)) - 2$。

因此，在二叉平衡树上进行查找时，查找过程中和给定值进行比较的关键字的个数和 *log(n)* 相当。