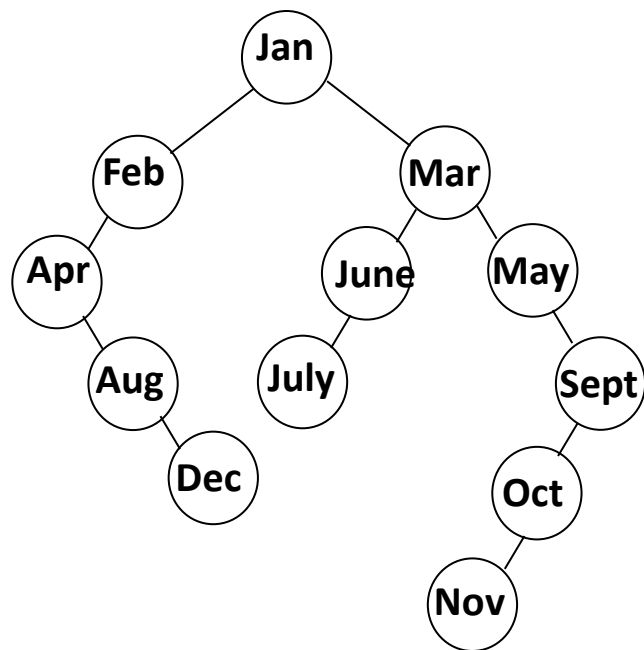


## 4.2 平衡二叉树

# 什么是平衡二叉树

【例】搜索树结点不同插入次序，将导致不同的深度和平均查找长度 **ASL**

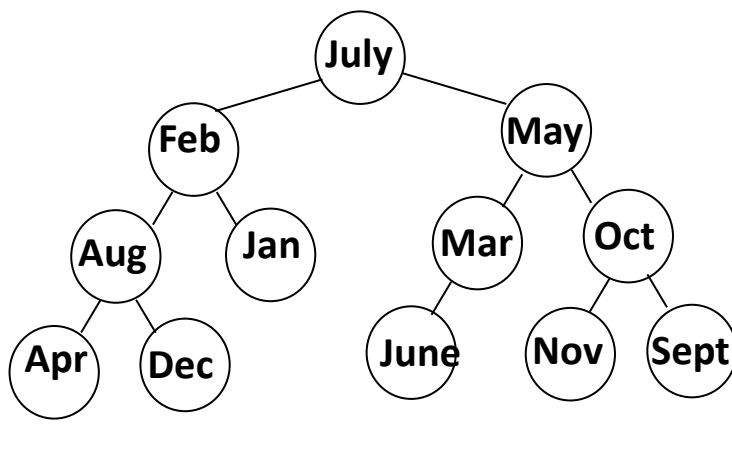
衡量查找效率（找一个值平均找几次）



(a) 自然月份序列

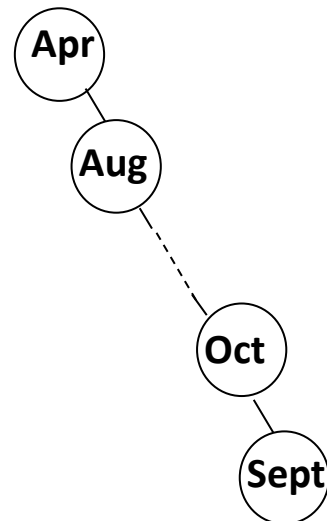
$$ASL(a) = (1 + 2 \times 2 + 3 \times 3 + 4 \times 3 + 5 \times 2 + 6 \times 1) / 12 = 3.5$$

1层的找1次，第二层找2次



(b) 按 July, Feb, May, Mar, Aug, Jan, Apr, Jun, Oct, Sept, Nov, Dec 人为指定顺序

$$ASL(b) = 3.0$$



(c) 月份字符串大小顺序

$$ASL(c) = 6.5$$

# 什么是平衡二叉树

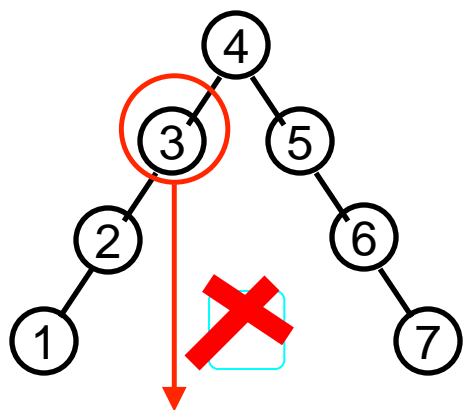
“平衡因子（Balance Factor，简称BF）： $BF(T) = h_L - h_R$ ，其中 $h_L$ 和 $h_R$ 分别为T的左、右子树的高度。

平衡是为了让树的高度更低一点

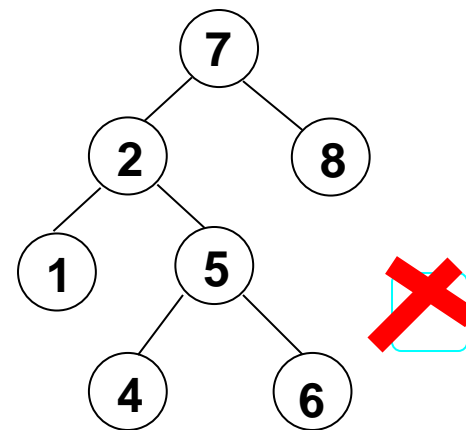
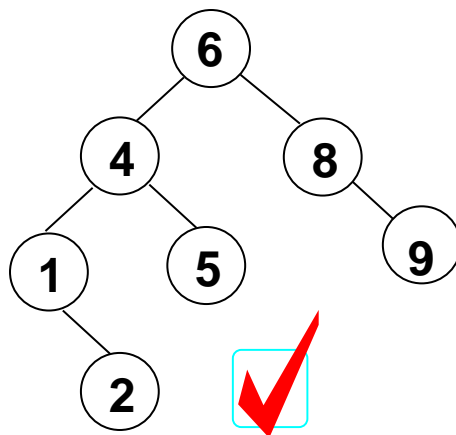
平衡二叉树（Balanced Binary Tree）（AVL树）

空树，或者

任一结点左、右子树高度差的绝对值不超过1，即 $|BF(T)| \leq 1$

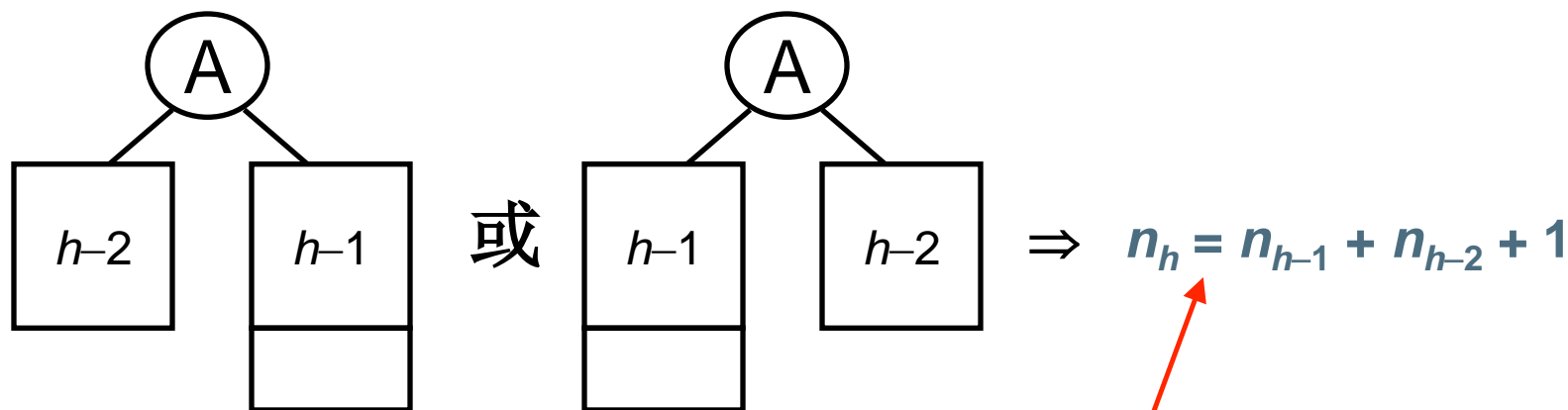


3左边高度为2，右边高度为0， $BF(3)=2$



平衡二叉树的高度能达到 $\log_2 n$ 吗？

设  $n_h$  高度为  $h$  的平衡二叉树的最少结点数。结点数最少时：



斐波那契序列：

$$F_0 = 1, F_1 = 1, F_i = F_{i-1} + F_{i-2} \text{ for } i > 1$$

设  $n_h$  是高度为  $h$  的平衡二叉树的最小结点数.

$h$	$n_h$	$F_h$
0	1	1
1	2	1
2	4	2
3	7	3
4	12	5
5	20	8
6	33	13
7	54	21
8	88	34
9	.....	

$$\Rightarrow n_h = n_{h-1} + n_{h-2} + 1$$

$$\Rightarrow n_h = F_{h+2} - 1, \quad (\text{对 } h \geq 0)$$

$$F_i \approx \frac{1}{\sqrt{5}} \left( \frac{1 + \sqrt{5}}{2} \right)^i$$

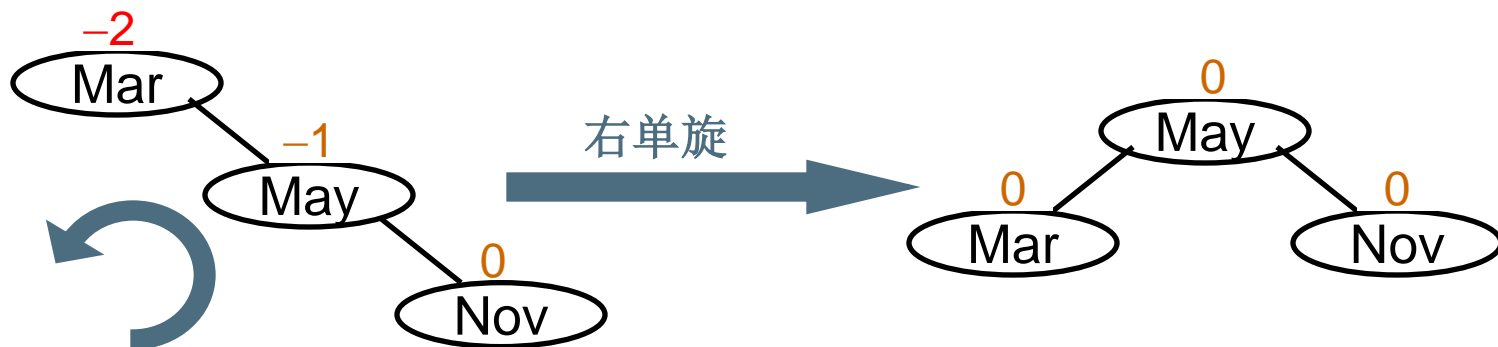
$$\Rightarrow n_h \approx \frac{1}{\sqrt{5}} \left( \frac{1 + \sqrt{5}}{2} \right)^{h+2} - 1$$

$$\Rightarrow h = O(\log_2 n) \quad \text{平衡二叉树}$$

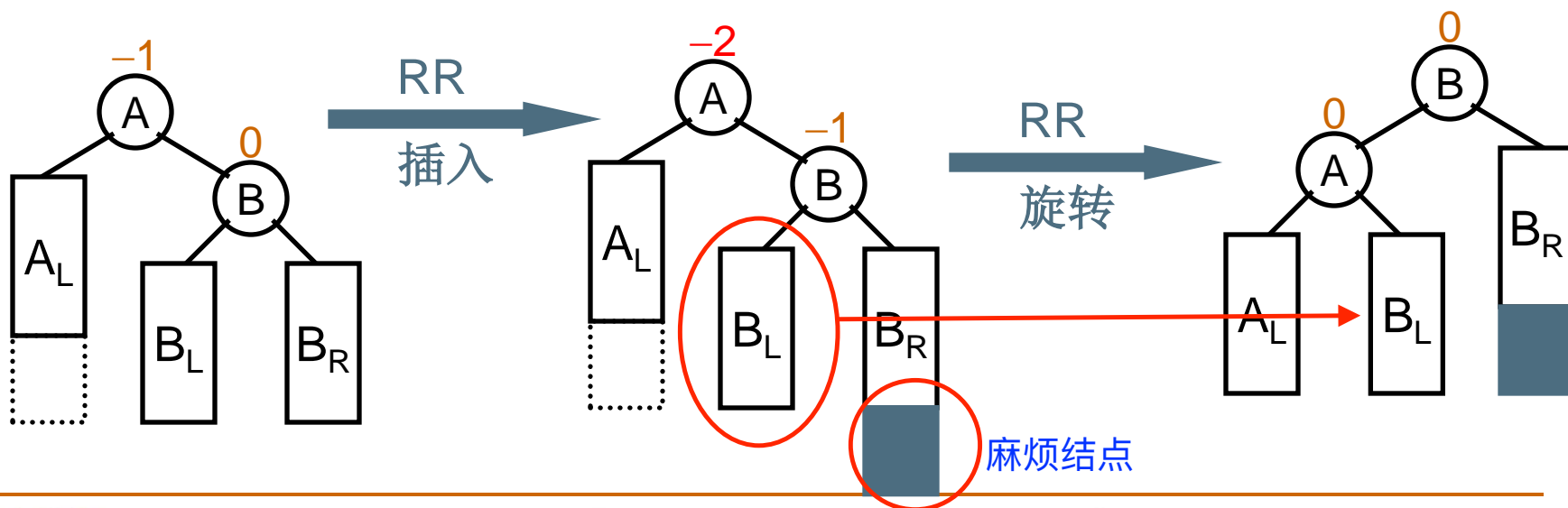
□ 给定结点数为  $n$  的AVL树的最大高度为  $O(\log_2 n)$ !

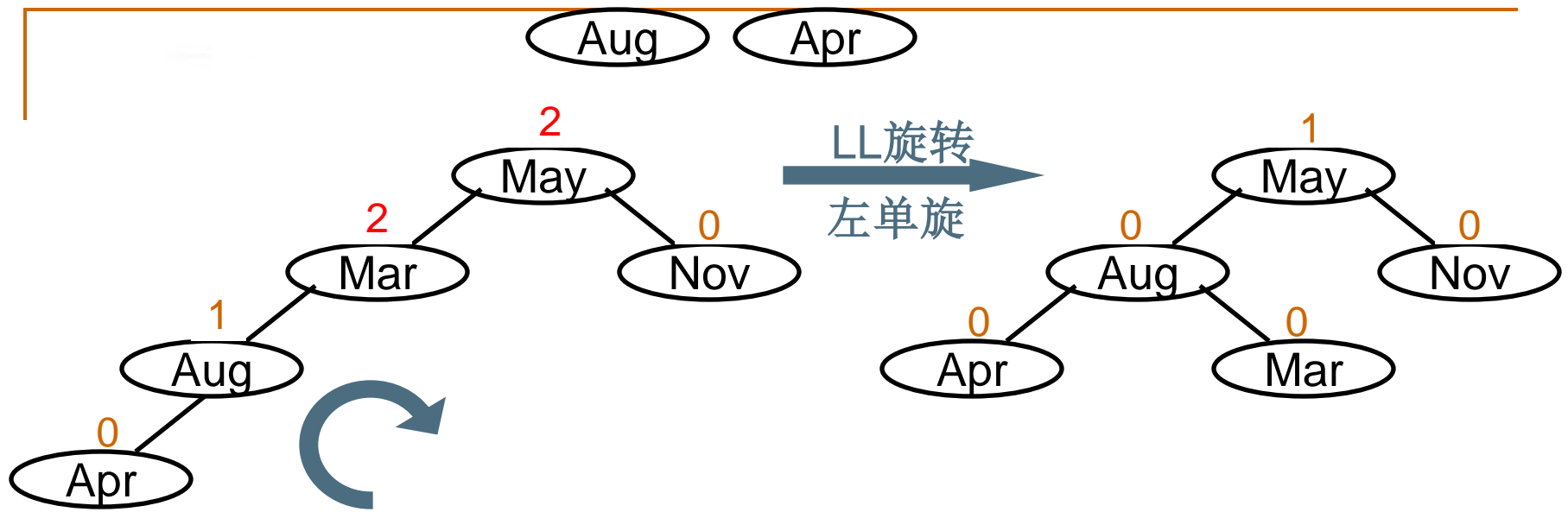
# 平衡二叉树的调整

# 平衡二叉树的调整

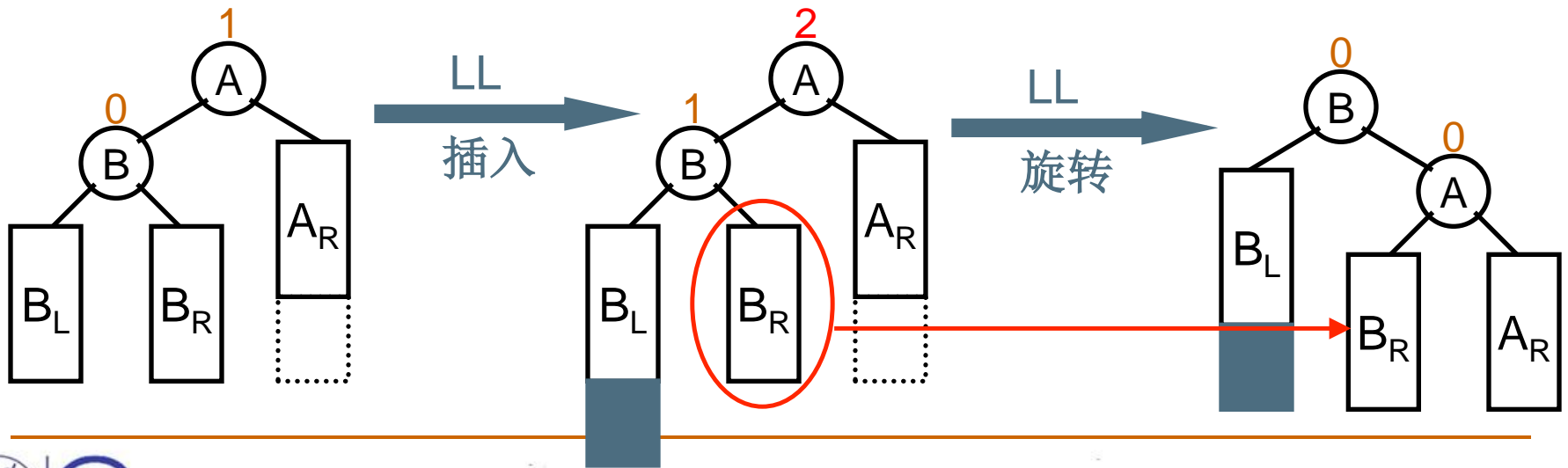


不平衡的“发现者”是Mar，“麻烦结点”Nov在发现者右子树的右边，因而叫RR插入，需要RR旋转（右单旋）

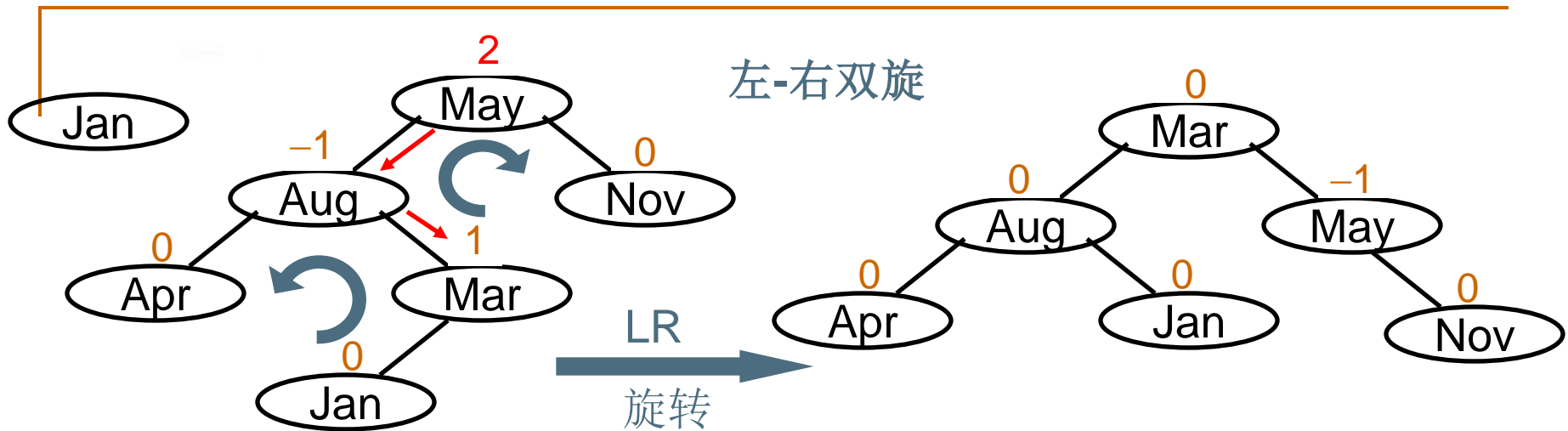




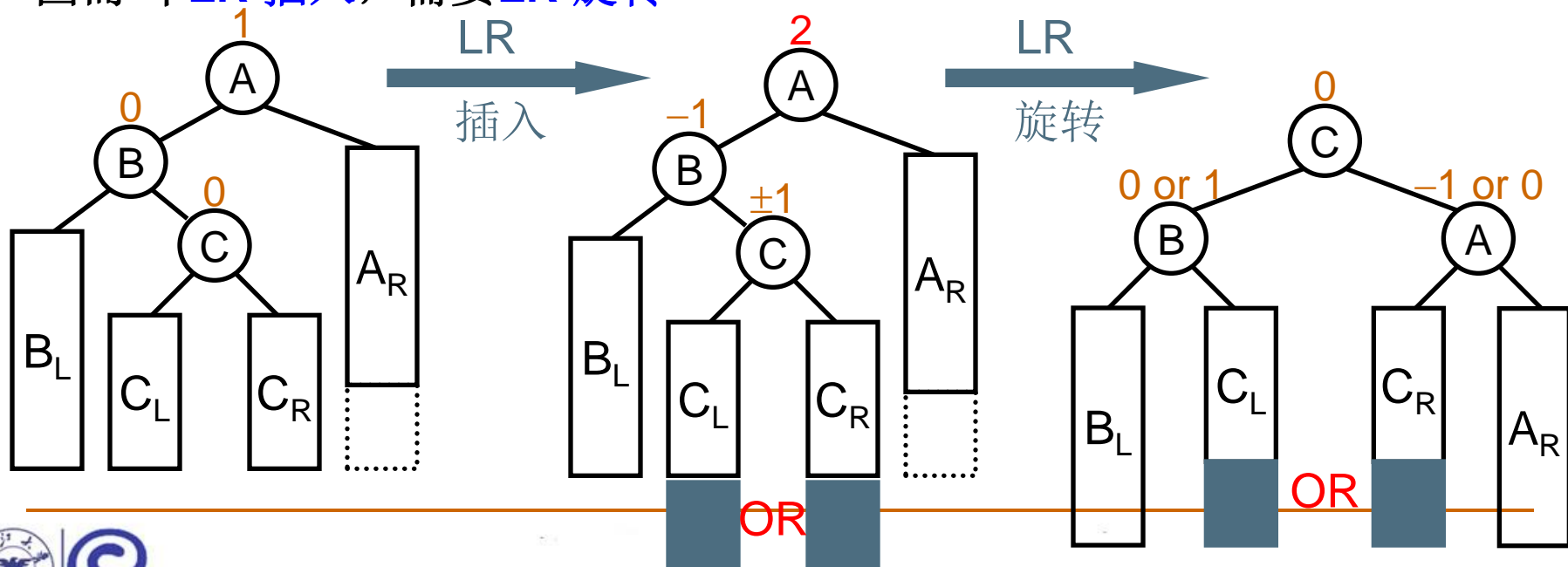
👉 “发现者”是**Mar**，“麻烦结点”**Apr**在发现者左子树的**左边**，因而叫 **LL 插入**，需要**LL 旋转（左单旋）**

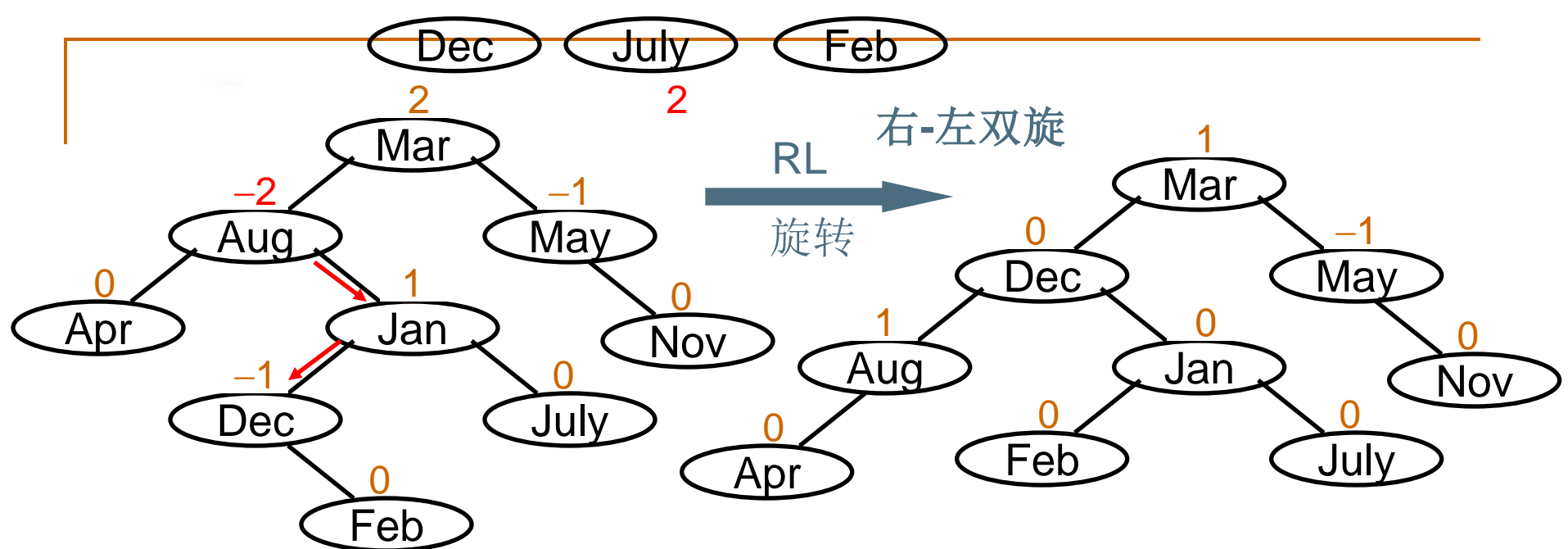




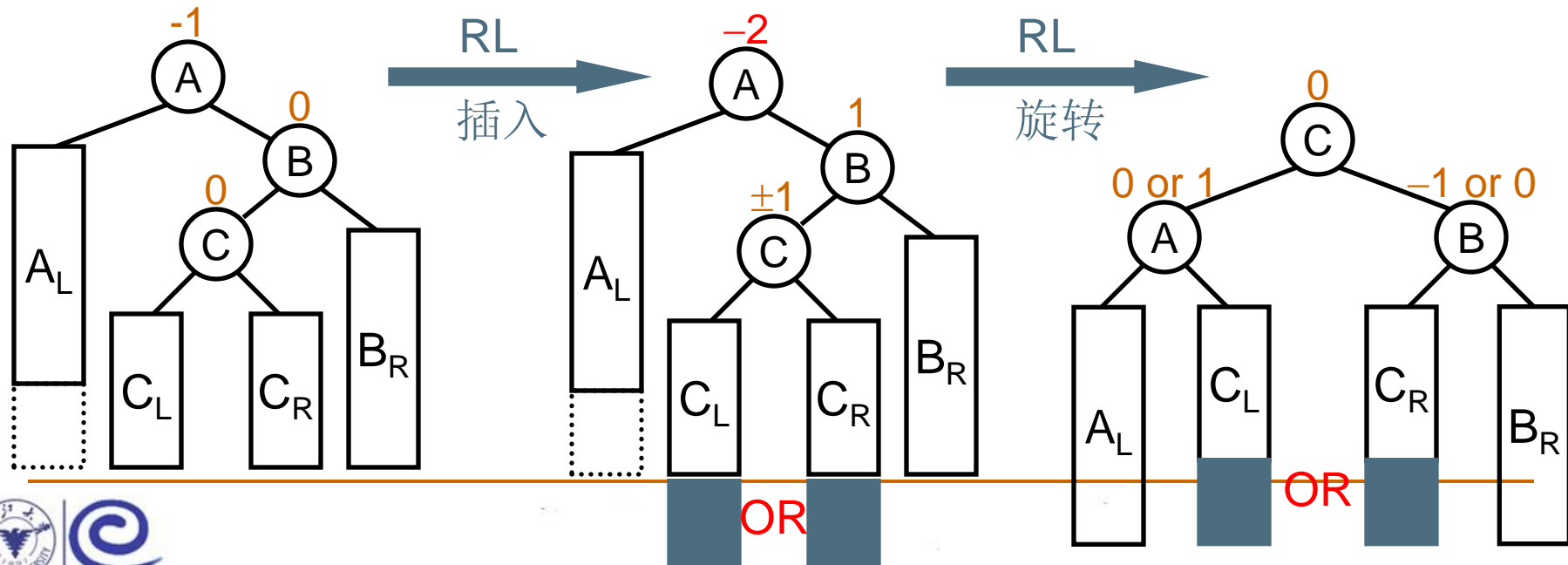


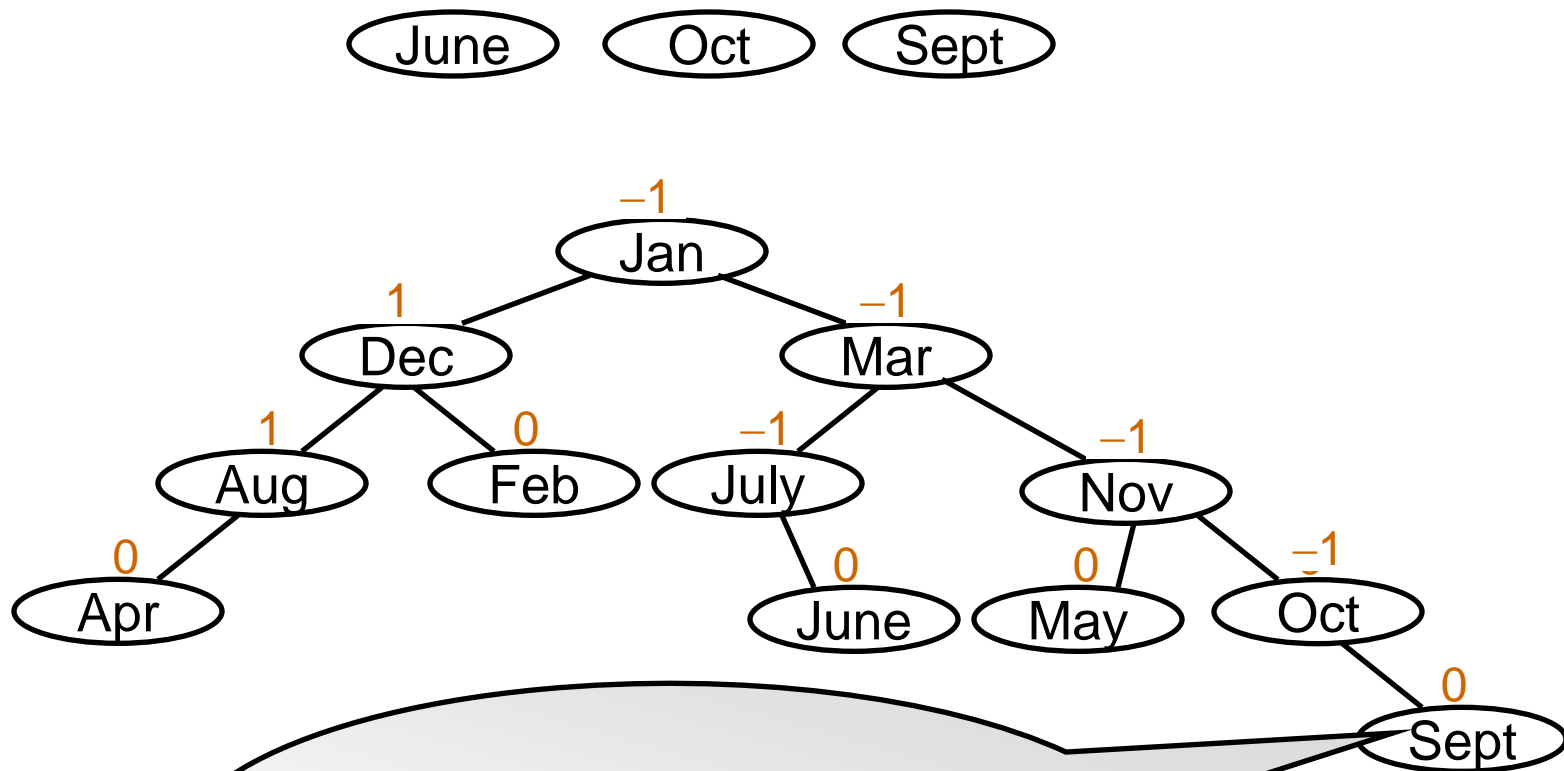
“发现者”是May，“麻烦结点”Jan在左子树的右边，因而叫 LR 插入，需要LR 旋转





👉 一般情况调整如下:





注意：有时候插入元素即便  
不需要调整结构，也可能需要重新计算  
一些平衡因子。