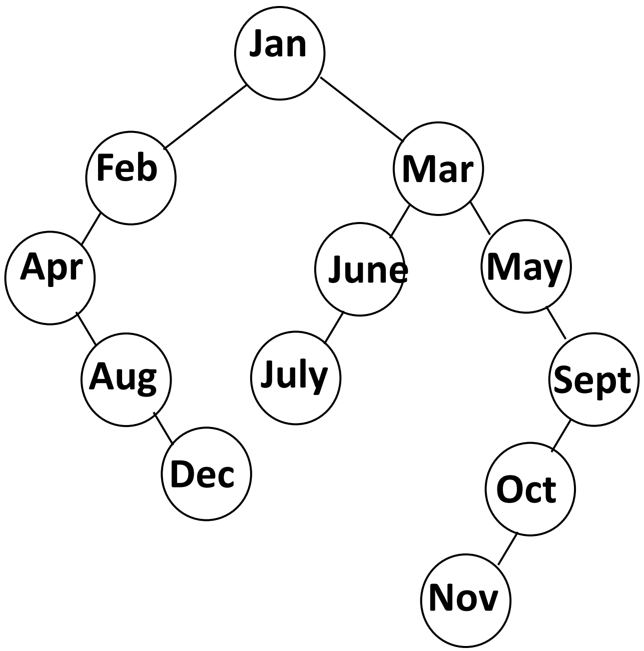
4.2平衡二叉树

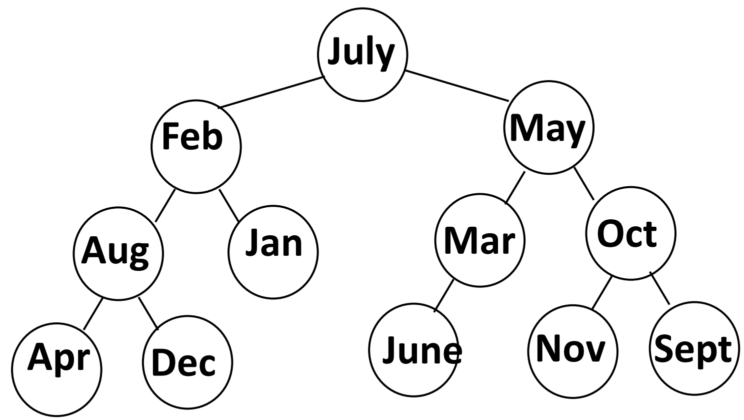
什么是平衡二叉树

〖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**



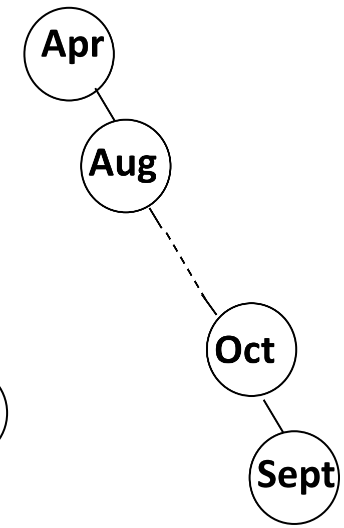
(a) 自然月份序列

ASL(a)=(1+2×2+3×3+4×3+5×2+6×1)/12 = 3.5



(b) 按**July, Feb, May, Mar, Aug, Jan, Apr, Jun, Oct, Sept, Nov, Dec**

ASL(b)=3.0



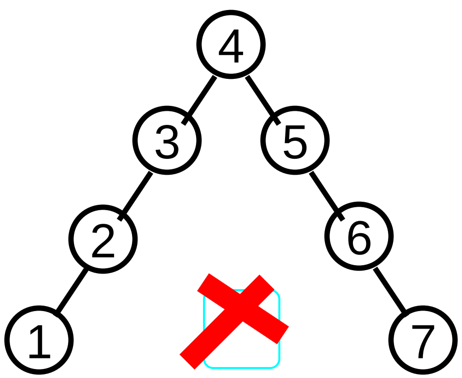
(c)月份字符串大小顺序

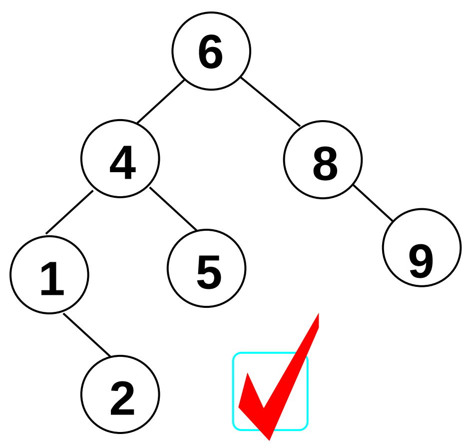
ASL(c)= 6.5

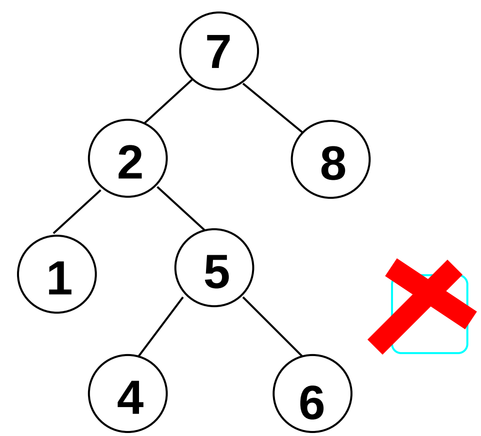
什么是平衡二叉树

“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$|BF(T)| \le 1$

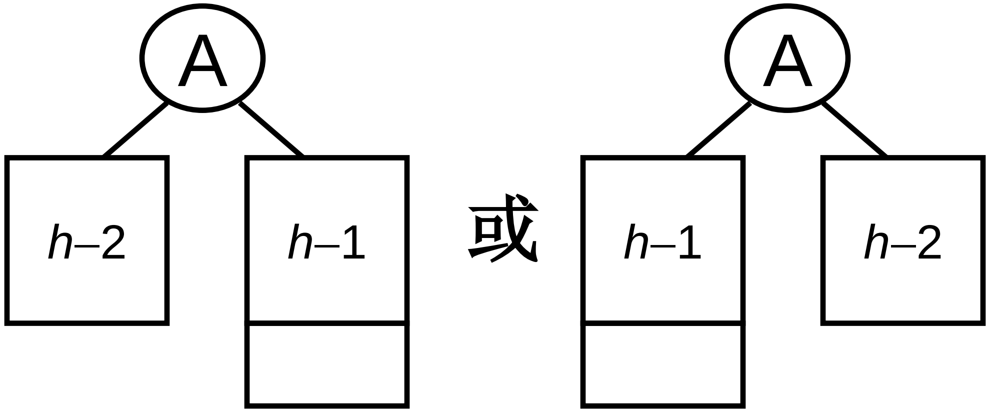






平衡二叉树的高度能达到$\log_2 n$吗？

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：



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

斐波那契序列:

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

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	…………………	

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

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

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

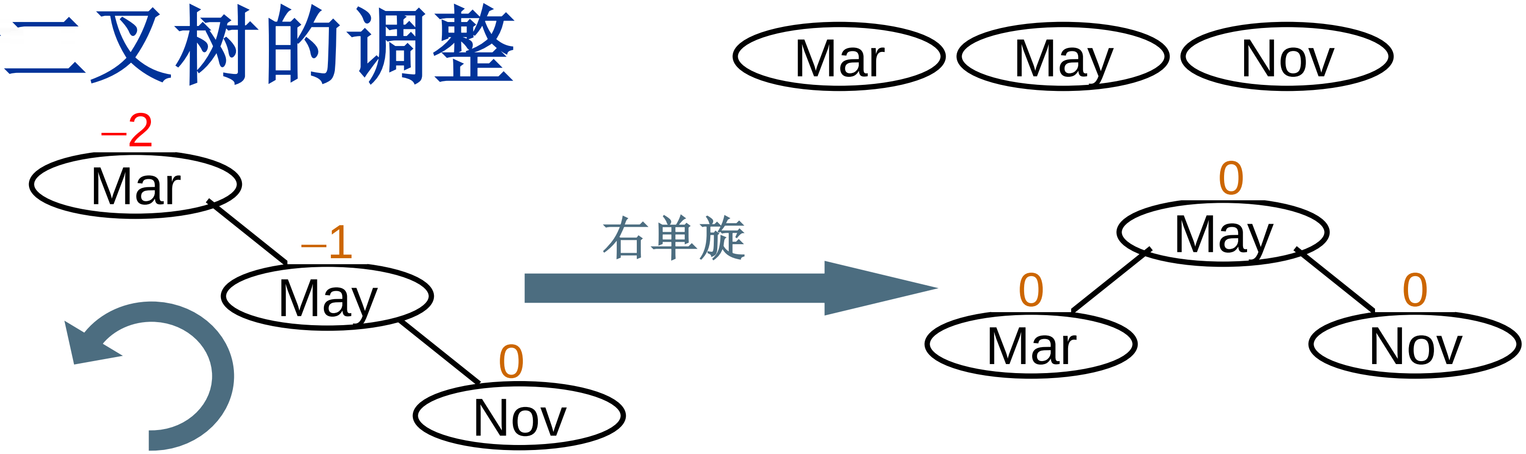
$$\Rightarrow \quad n_h \approx \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{h+2} - 1$$

$$\Rightarrow \quad h = O(\log_2 n)$$

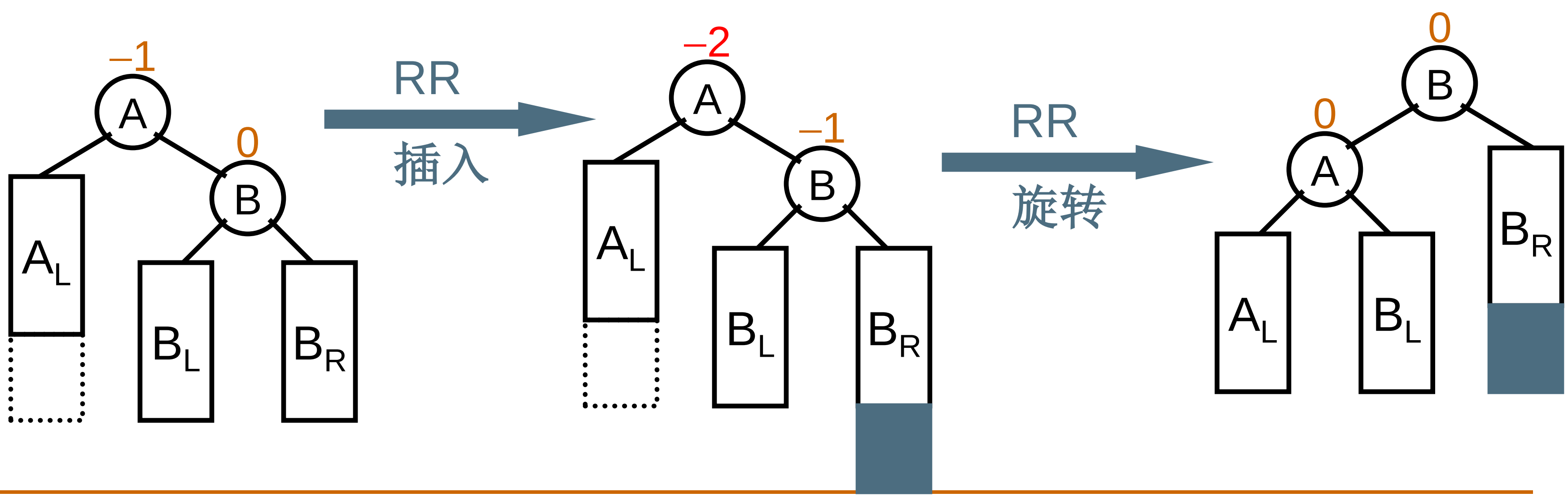
□ 给定结点数为 n的AVL树的最大高度为$O(\log_2 n)$！

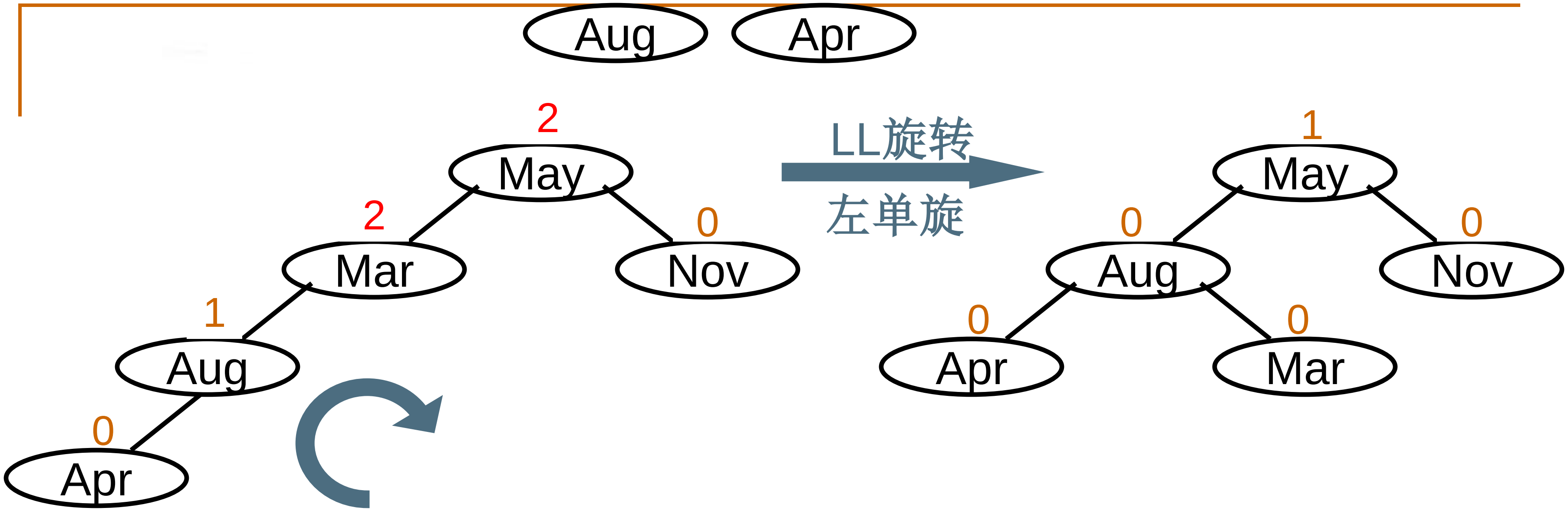
平衡二叉树的调整

平衡二叉树的调整

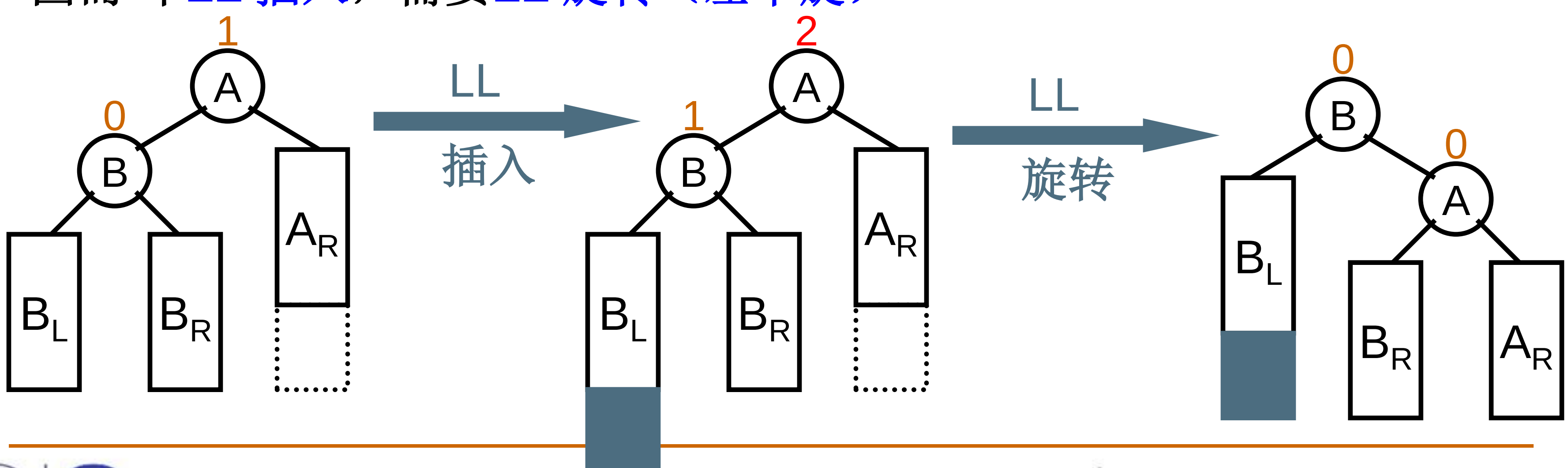


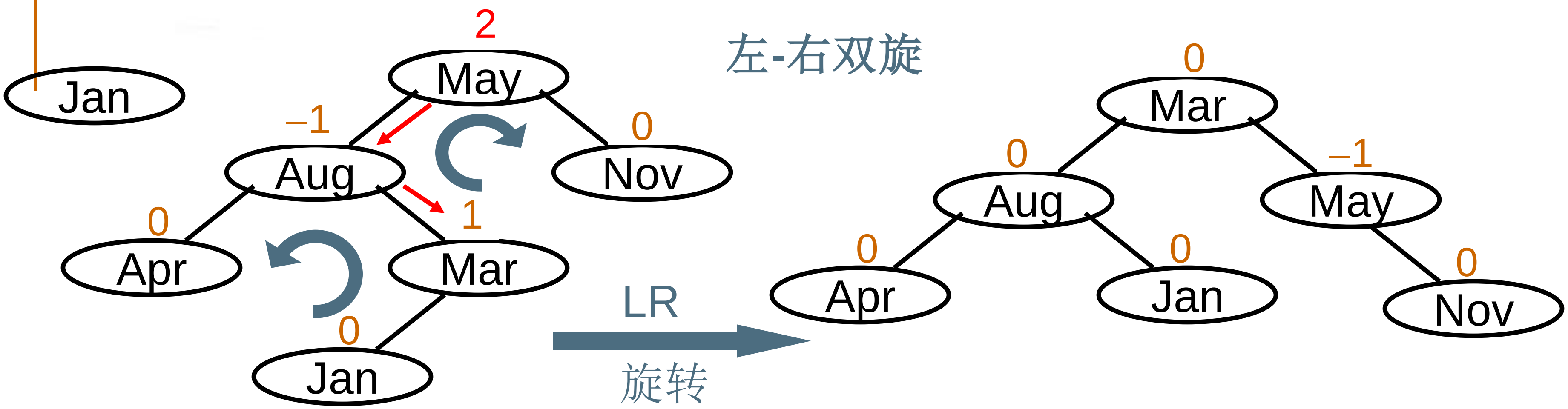
不平衡的“**发现者**”是**Mar**，“**麻烦结点**”**Nov** 在发现者**右**子树的**右边**，因而叫 **RR 插入**，需要**RR 旋转（右单旋）**



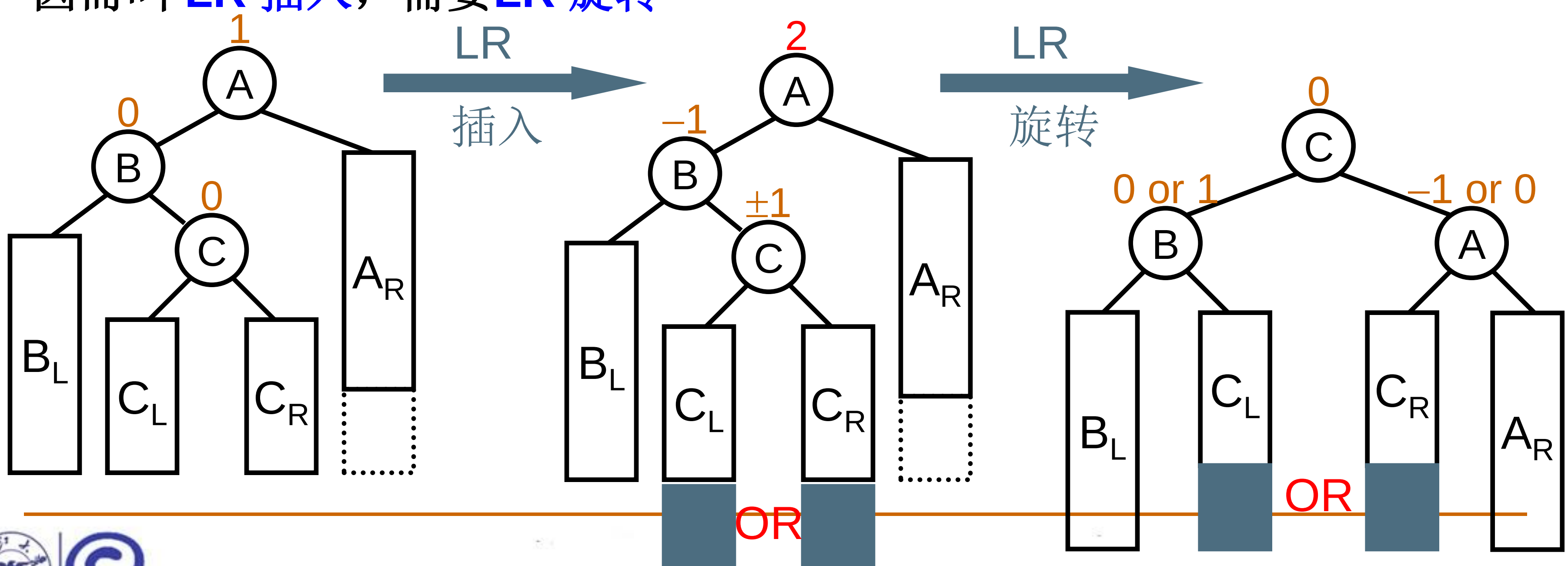


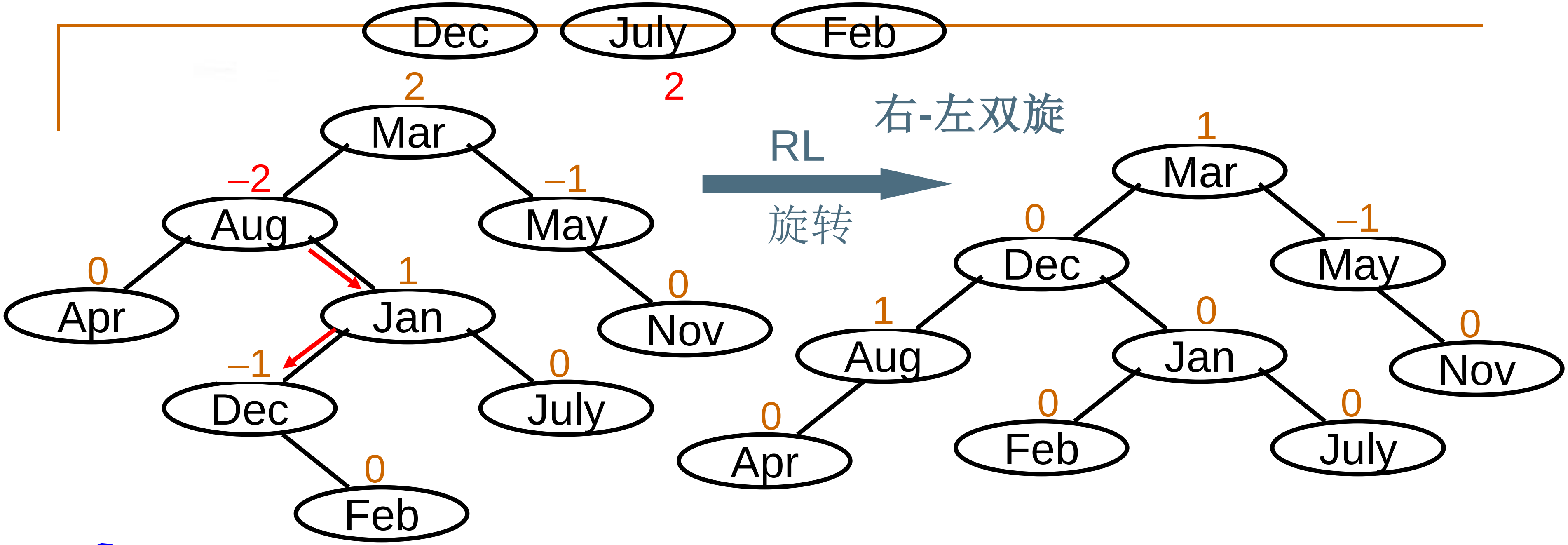
“发现者”是**Mar**， “麻烦结点”**Apr** 在发现者左子树的左边，因而叫 **LL** 插入，需要**LL** 旋转（左单旋）



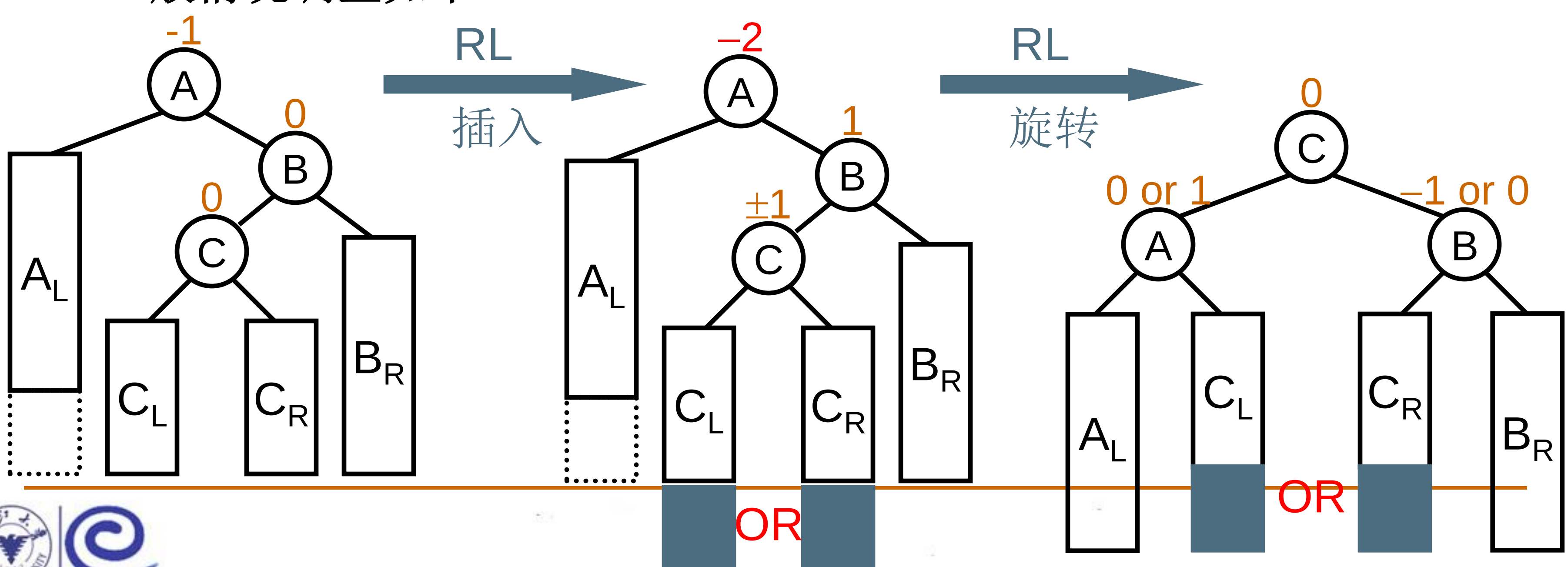


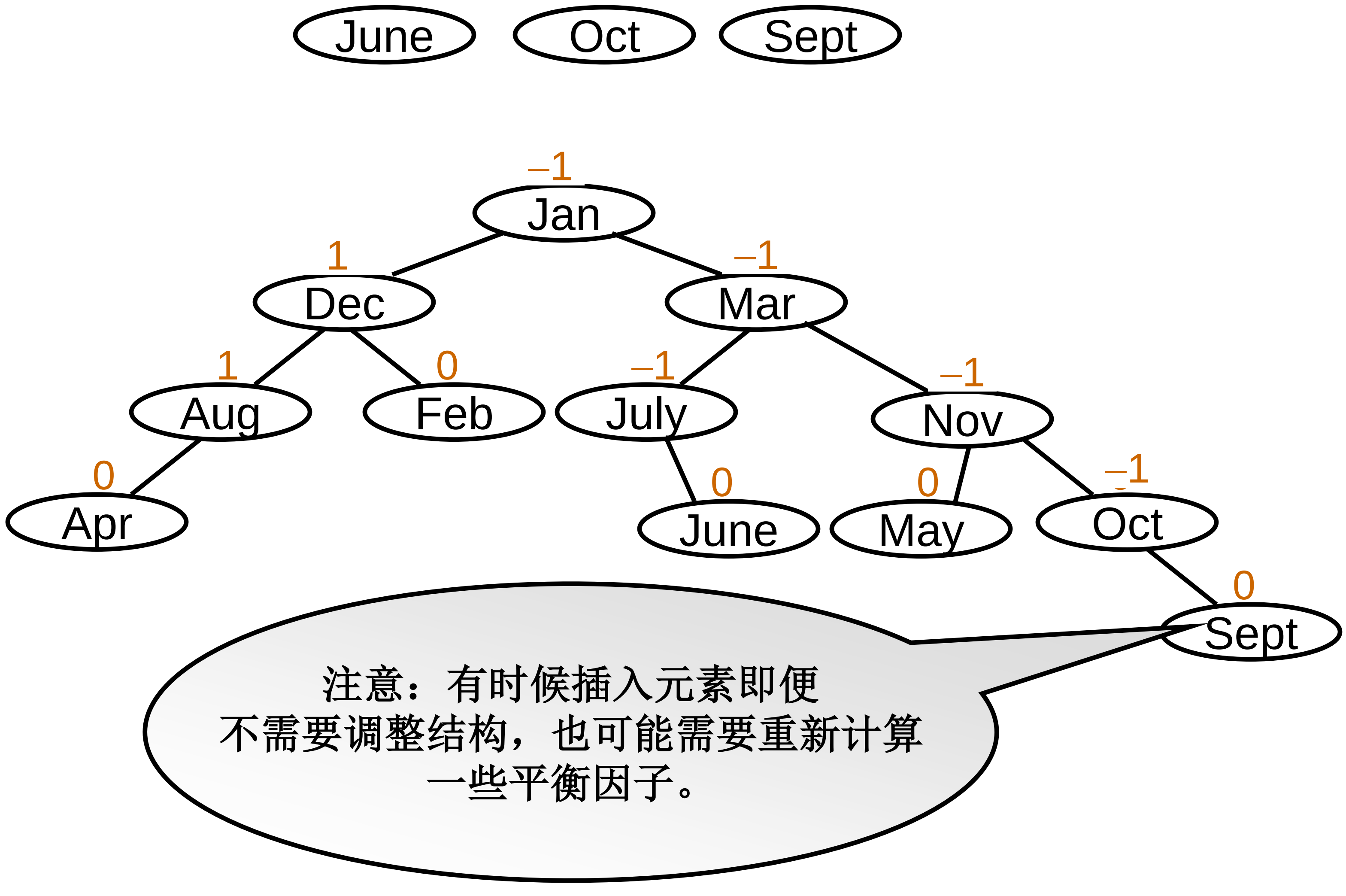
☞ “**发现者**”是**May**，“**麻烦结点**”**Jan**在**左**子树的**右边**，因而叫 **LR 插入**，需要**LR 旋转**




Dec
July
Feb
右-左双旋
RL
旋转
Mar
Aug
May
Apr
Jan
Nov
Dec
July
Feb

一般情况调整如下:


RL
插入
RL
旋转
A
B
C
A_L
C_L
C_R
B_R
0 or 1
–1 or 0
OR
OR


June
Oct
Sept
–1
Jan
1
Dec
–1
Mar
1
Aug
0
Feb
–1
July
–1
Nov
0
Apr
0
June
0
May
–1
Oct
0
Sept
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。