本节内容

红黑树

插入操作

红黑树的插入

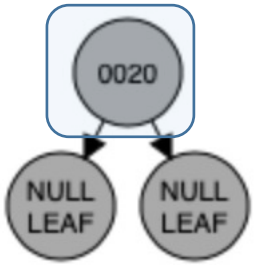
从一棵空的红黑树开始，插入：20, 10, 5, 30, 40, 57, 3, 2, 4, 35, 25, 18, 22, 23, 24, 19, 18

- 先查找，确定插入位置（原理同二叉排序树），插入新结点
- 新结点是根——染为黑色
- 新结点非根——染为红色
  - 若插入新结点后依然满足红黑树定义，则插入结束
  - 若插入新结点后不满足红黑树定义，需要调整，使其重新满足红黑树定义
    - 黑叔：旋转+染色
      - LL型：右单旋，父换爷+染色
      - RR型：左单旋，父换爷+染色
      - LR型：左、右双旋，儿换爷+染色
      - RL型：右、左双旋，儿换爷+染色
    - 红叔：染色+变新
      - 叔父爷染色，爷变为新结点

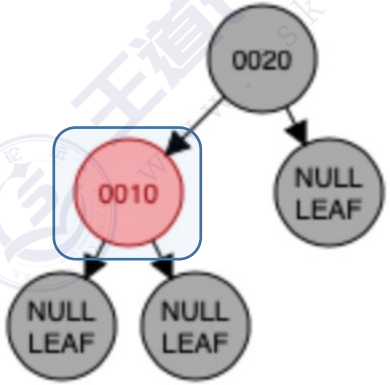
如何调整：看新结点叔叔的脸色

红黑树的插入

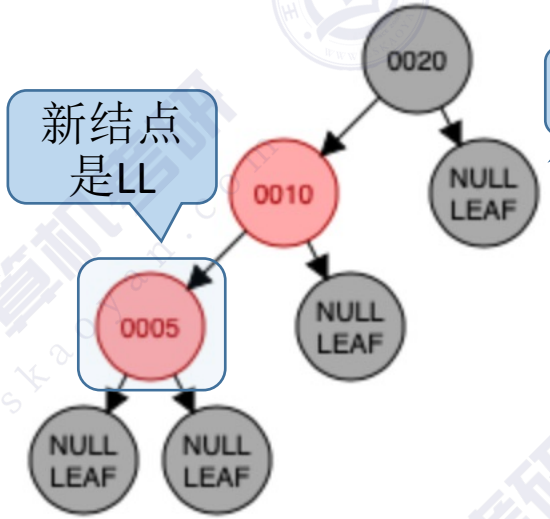
从一棵空的红黑树开始，插入：20, 10, 5, 30, 40, 57, 3, 2, 4, 35, 25, 18, 22, 23, 24, 19, 18



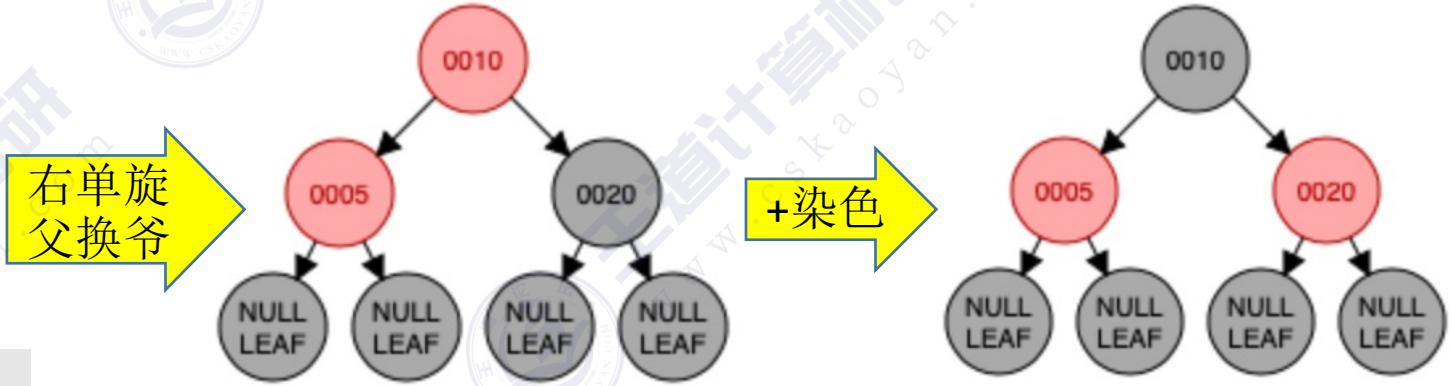
新结点是根，染黑



新结点非根，染红
依然满足红黑树



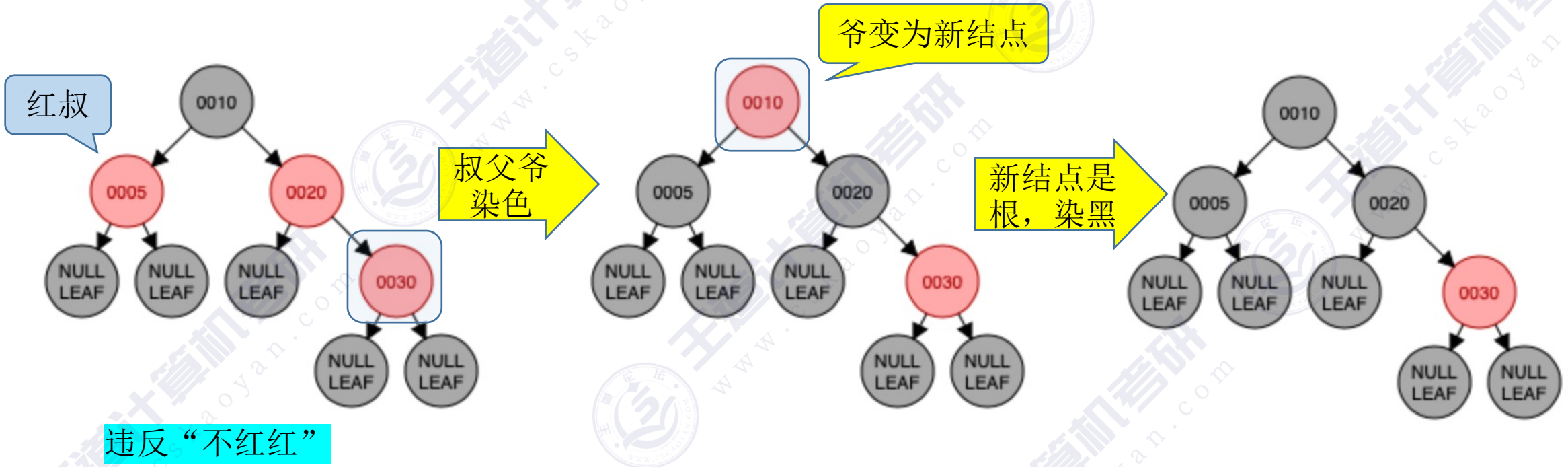
新结点非根，染红
违反“不红红”



- 黑叔：旋转+染色
  - LL型：右单旋，父换爷+染色

红黑树的插入

从一棵空的红黑树开始，插入：20, 10, 5, 30, 40, 57, 3, 2, 4, 35, 25, 18, 22, 23, 24, 19, 18

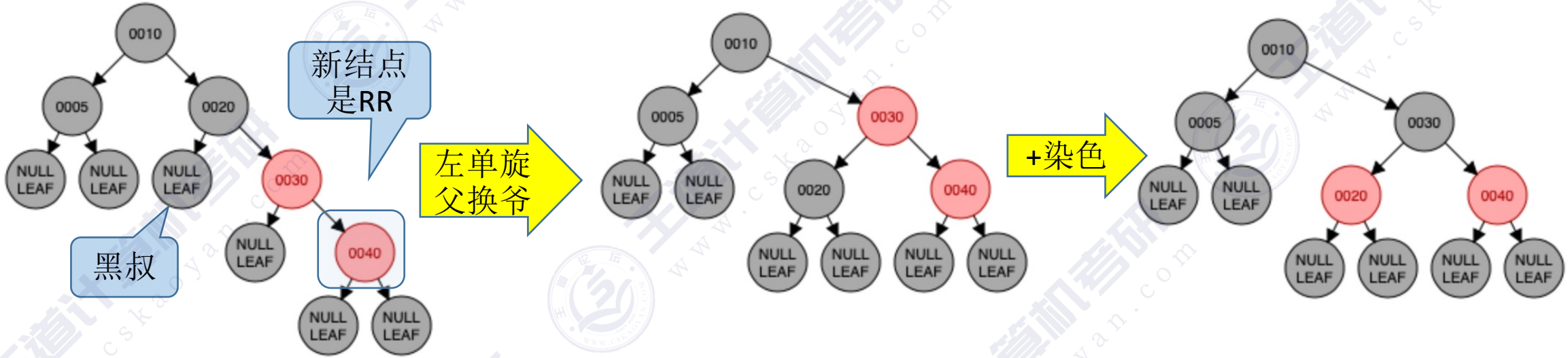


违反“不红红”

- 红叔：染色+变新
  - 叔父爷染色，爷变为新结点

红黑树的插入

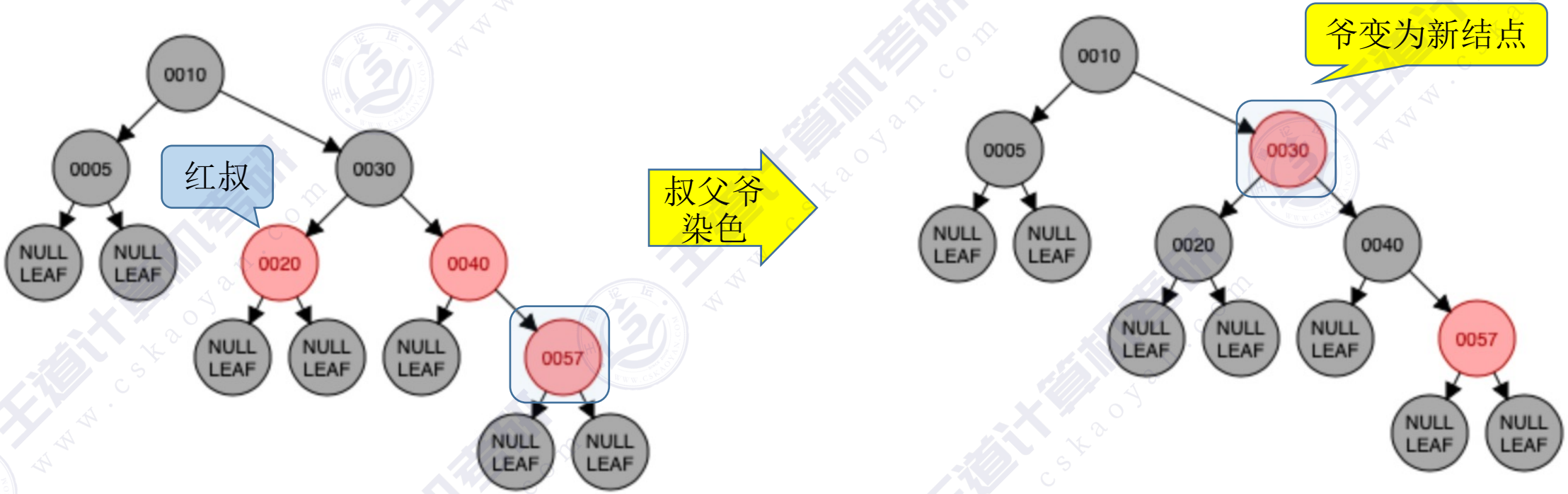
从一棵空的红黑树开始，插入：20, 10, 5, 30, 40, 57, 3, 2, 4, 35, 25, 18, 22, 23, 24, 19, 18



- 黑叔：旋转+染色
  - RR型：左单旋，父换爷+染色

红黑树的插入

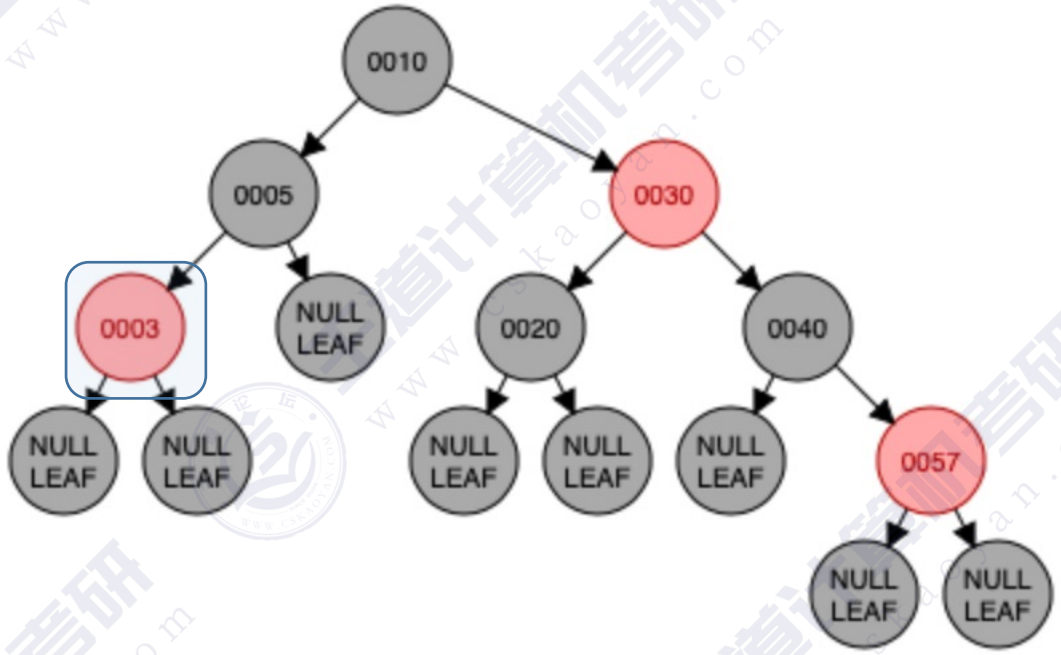
从一棵空的红黑树开始，插入：20, 10, 5, 30, 40, 57, 3, 2, 4, 35, 25, 18, 22, 23, 24, 19, 18



- 红叔：染色+变新
  - 叔父爷染色，爷变为新结点

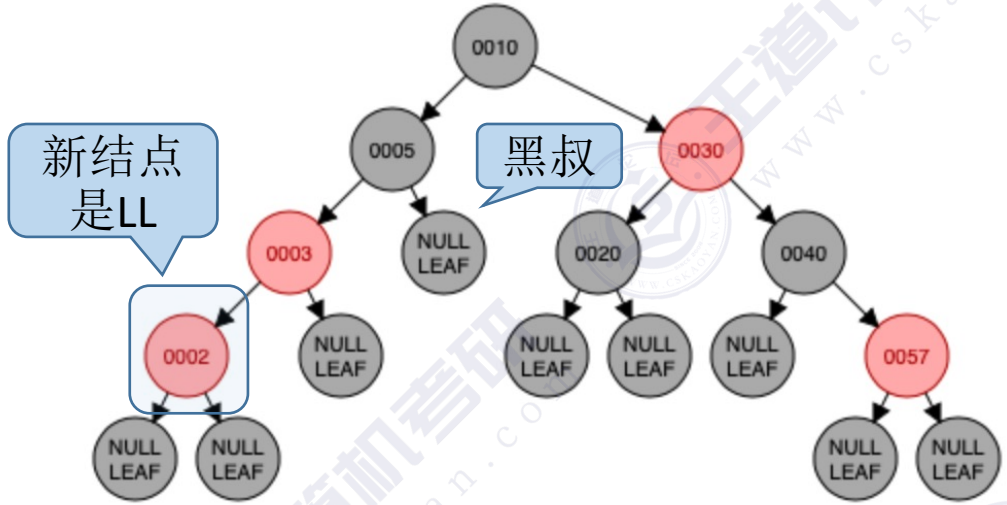
红黑树的插入

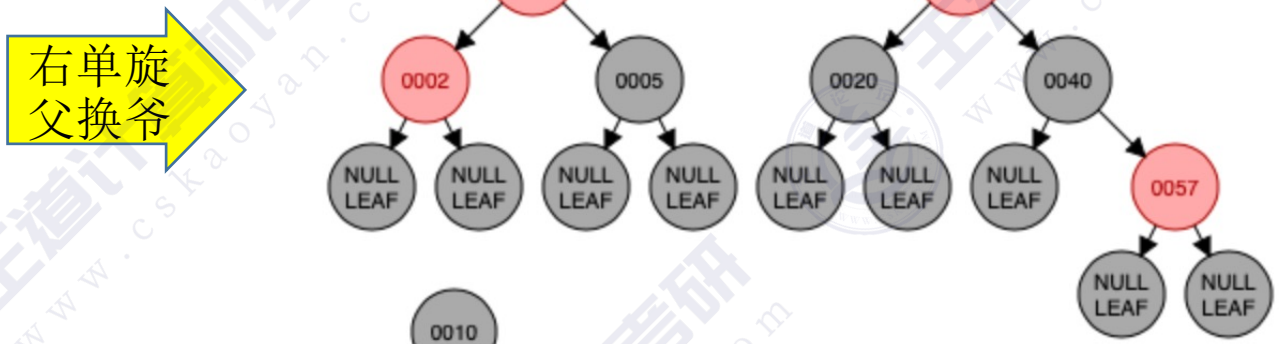
从一棵空的红黑树开始，插入：20, 10, 5, 30, 40, 57, 3, 2, 4, 35, 25, 18, 22, 23, 24, 19, 18

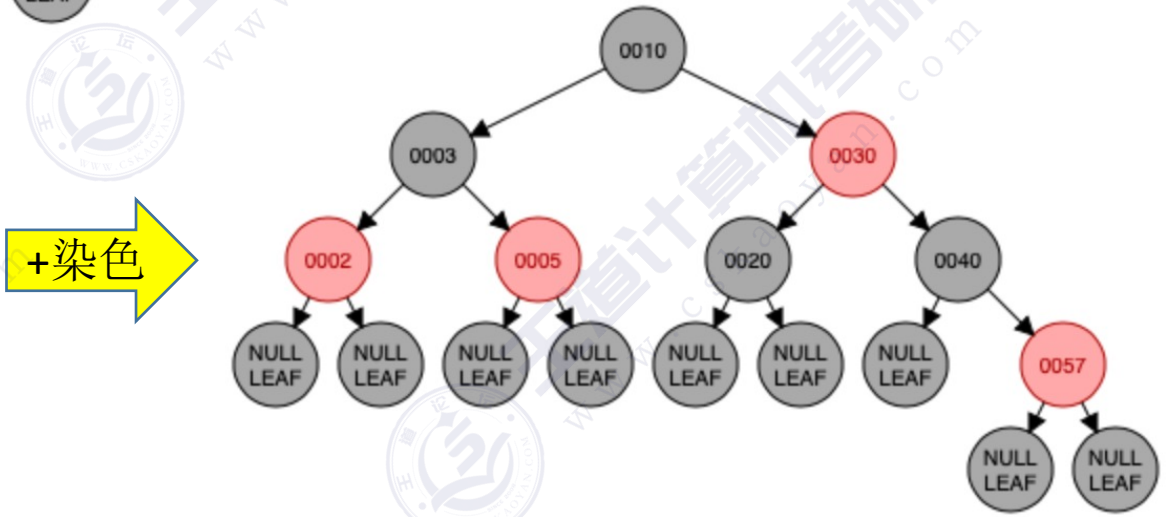


红黑树的插入

从一棵空的红黑树开始，插入：20, 10, 5, 30, 40, 57, 3, 2, 4, 35, 25, 18, 22, 23, 24, 19, 18



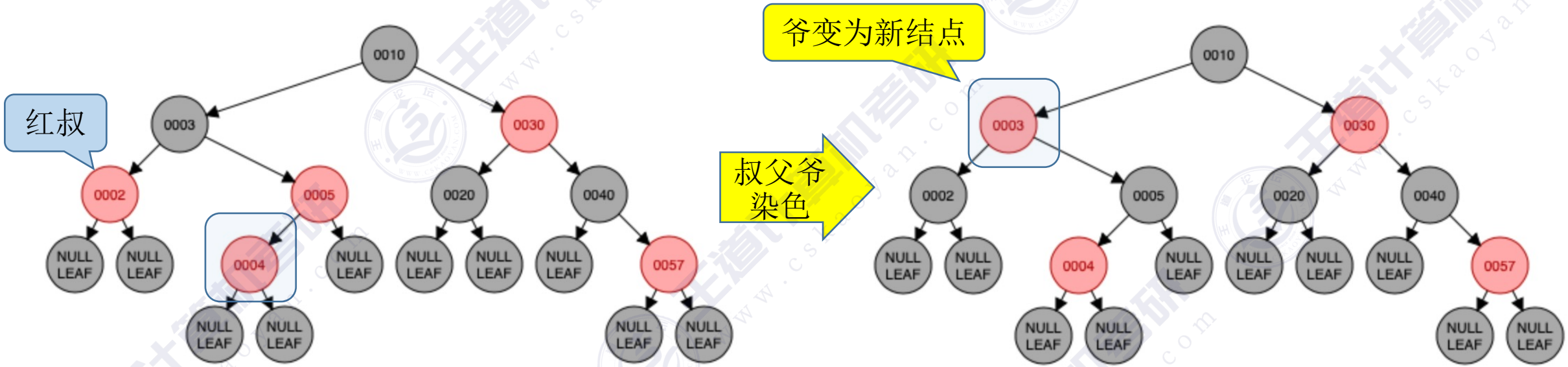




- 黑叔：旋转+染色
  - LL型：右单旋，父换爷+染色

红黑树的插入

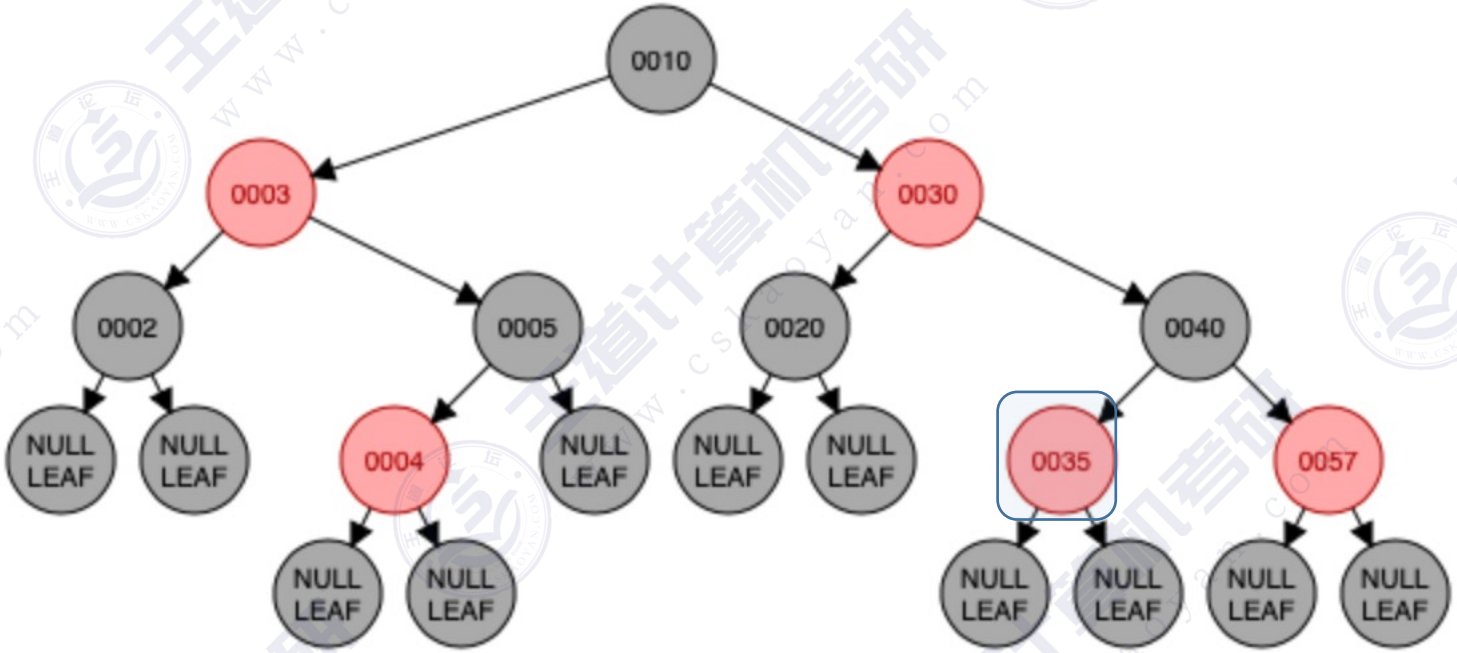
从一棵空的红黑树开始，插入：20, 10, 5, 30, 40, 57, 3, 2, 4, 35, 25, 18, 22, 23, 24, 19, 18



- 红叔：染色+变新
  - 叔父爷染色，爷变为新结点

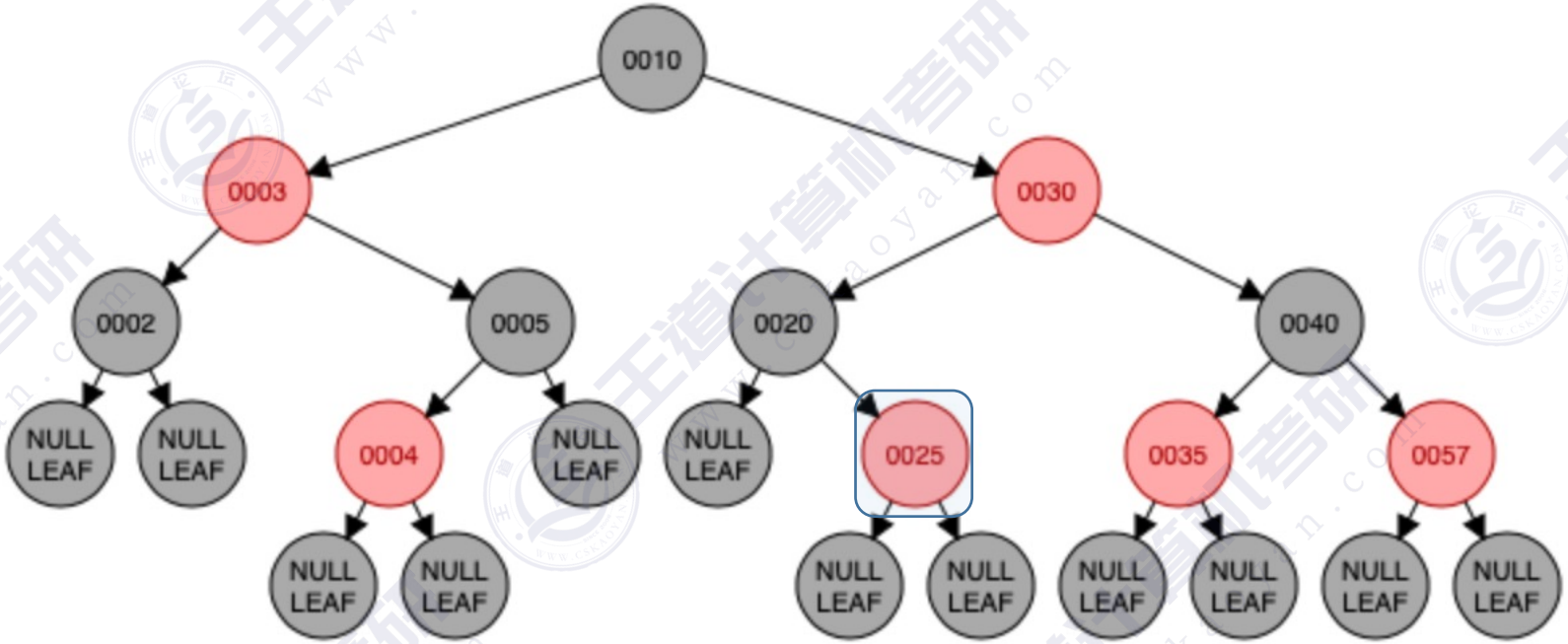
红黑树的插入

从一棵空的红黑树开始，插入：20, 10, 5, 30, 40, 57, 3, 2, 4, 35, 25, 18, 22, 23, 24, 19, 18



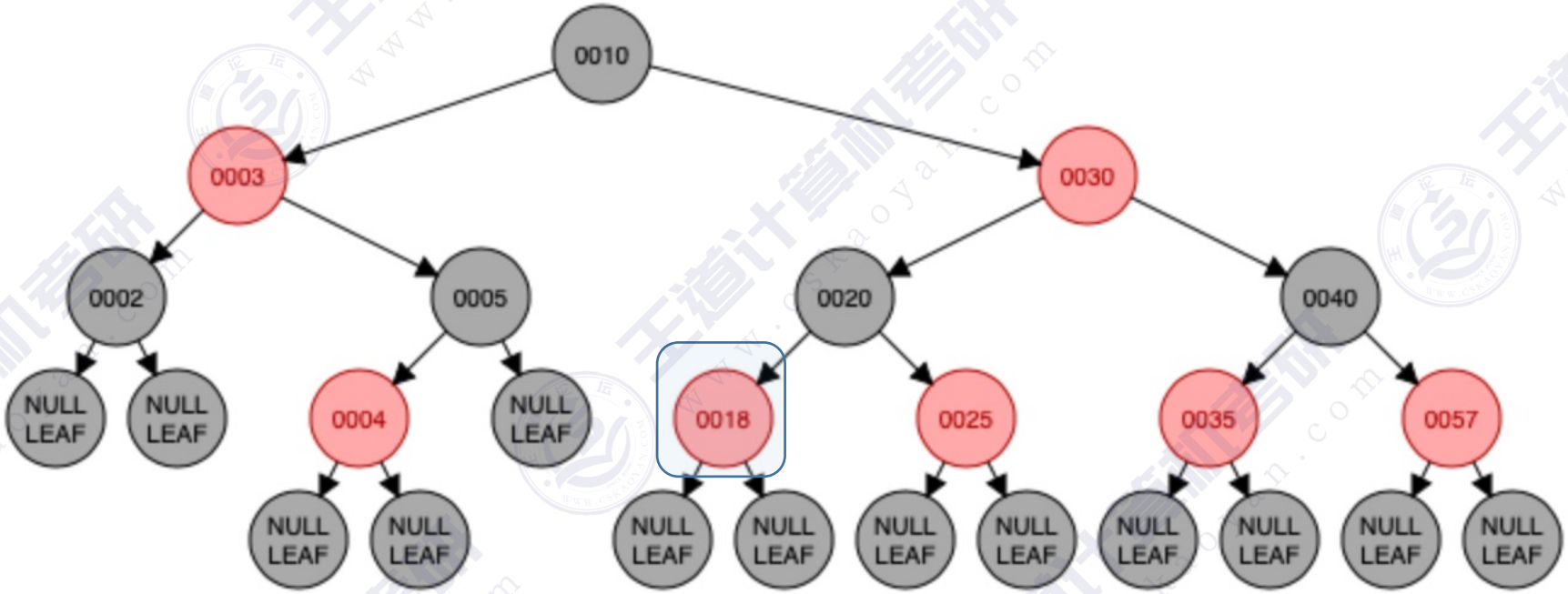
红黑树的插入

从一棵空的红黑树开始，插入：20, 10, 5, 30, 40, 57, 3, 2, 4, 35, 25, 18, 22, 23, 24, 19, 18



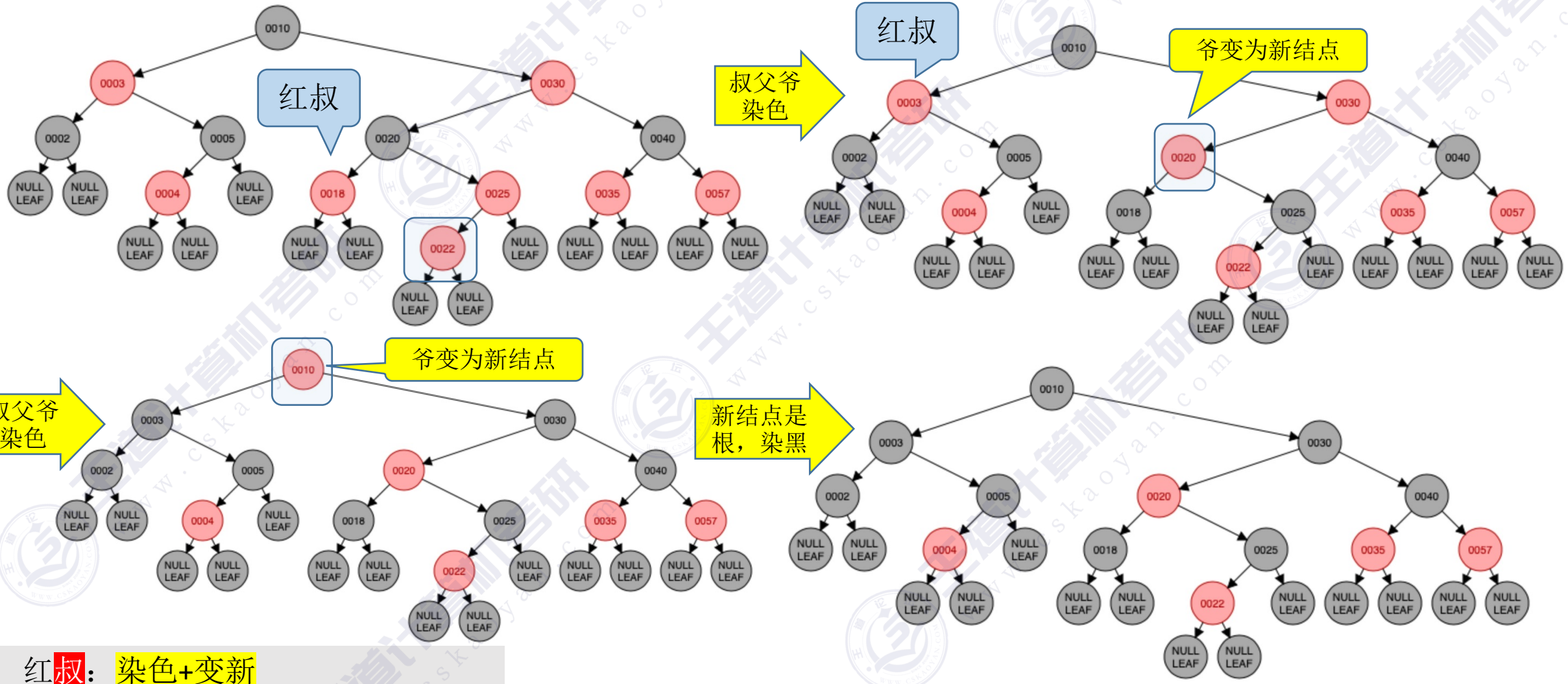
红黑树的插入

从一棵空的红黑树开始，插入：20, 10, 5, 30, 40, 57, 3, 2, 4, 35, 25, 18, 22, 23, 24, 19, 18



红黑树的插入

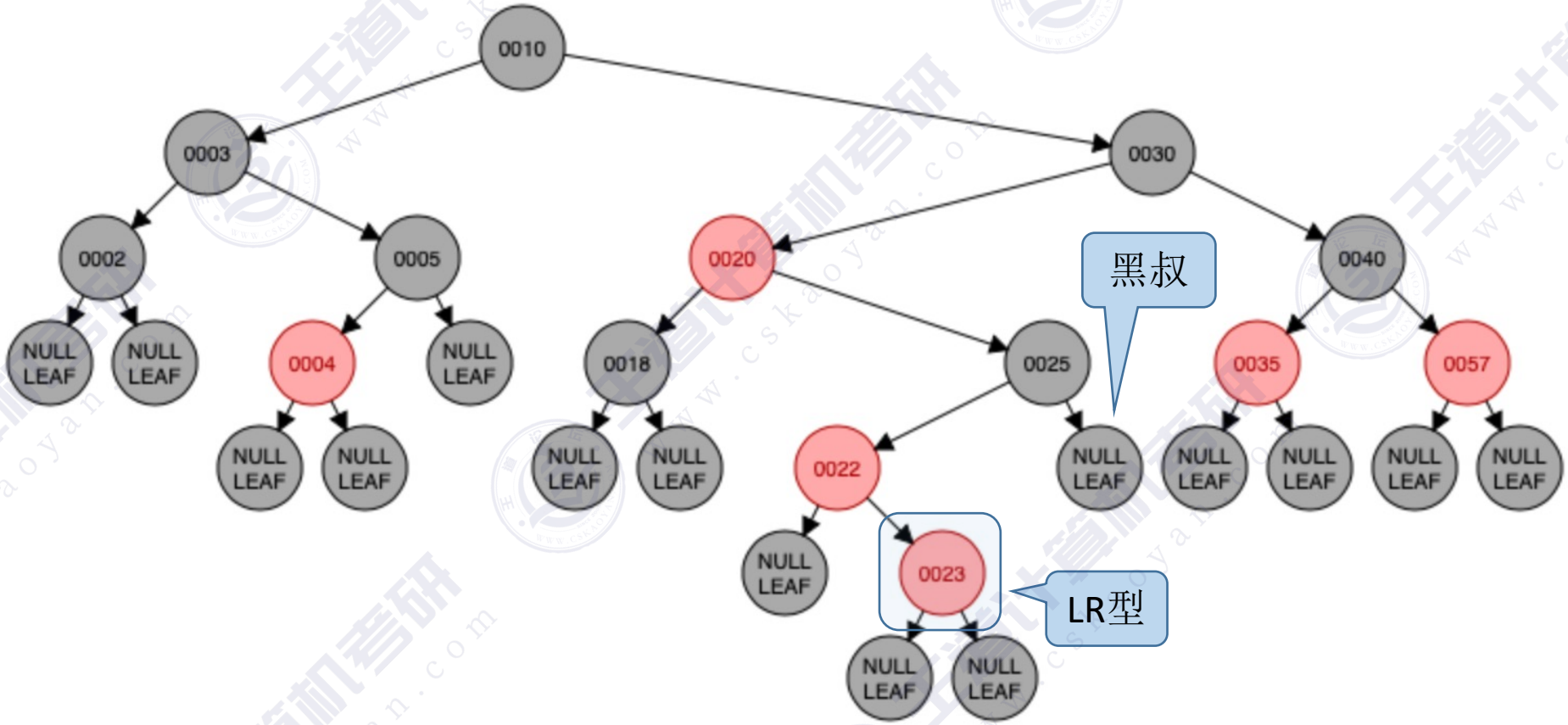
从一棵空的红黑树开始，插入：20, 10, 5, 30, 40, 57, 3, 2, 4, 35, 25, 18, 22, 23, 24, 19, 18



- 红叔：染色+变新
  - 叔父爷染色，爷变为新结点

红黑树的插入

从一棵空的红黑树开始，插入：20, 10, 5, 30, 40, 57, 3, 2, 4, 35, 25, 18, 22, 23, 24, 19, 18

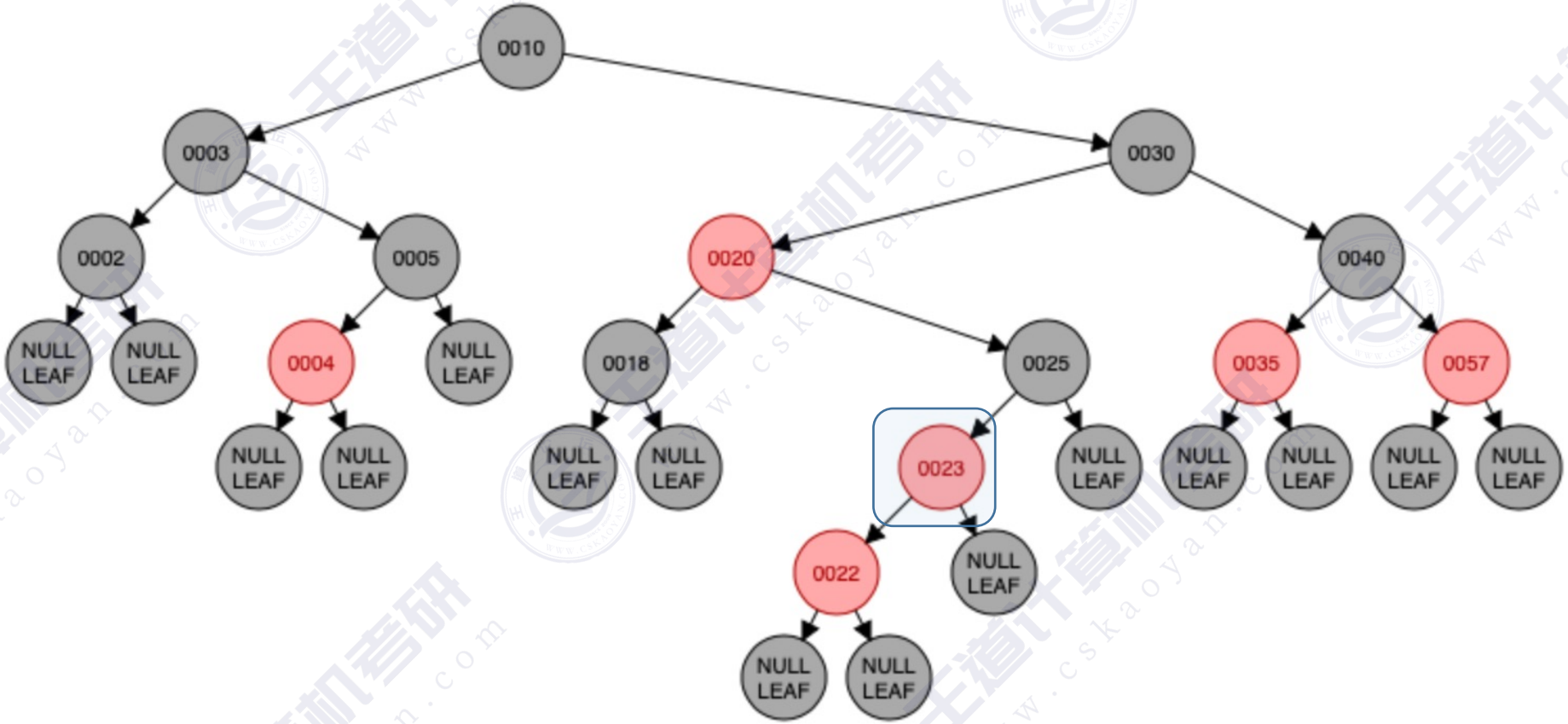


- 黑叔：旋转+染色
  - LR型：左、右双旋，儿换爷+染色

红黑树的插入

从一棵空的红黑树开始，插入：20, 10, 5, 30, 40, 57, 3, 2, 4, 35, 25, 18, 22, 23, 24, 19, 18

左单旋



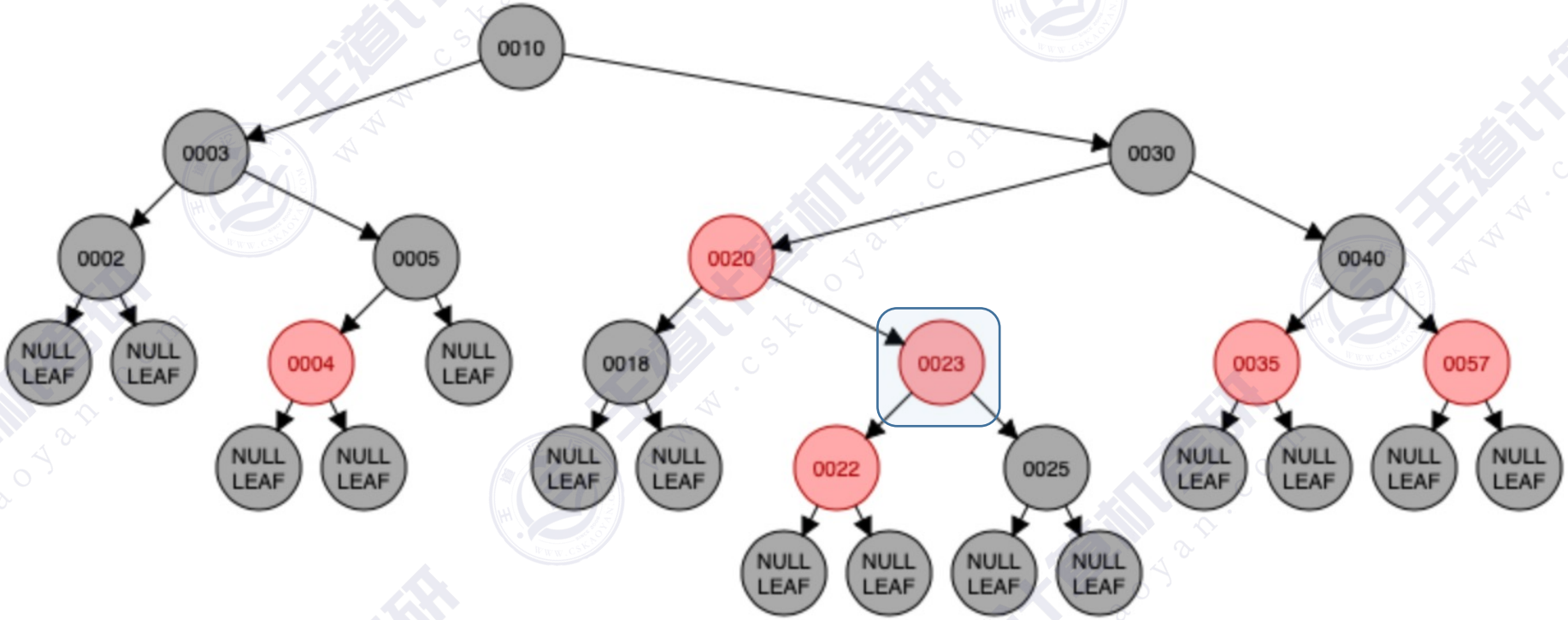
①左单旋的结果：儿子辈分升高

- 黑叔：旋转+染色
  - LR型：左、右双旋，儿换爷+染色

红黑树的插入

从一棵空的红黑树开始，插入：20, 10, 5, 30, 40, 57, 3, 2, 4, 35, 25, 18, 22, 23, 24, 19, 18

右单旋



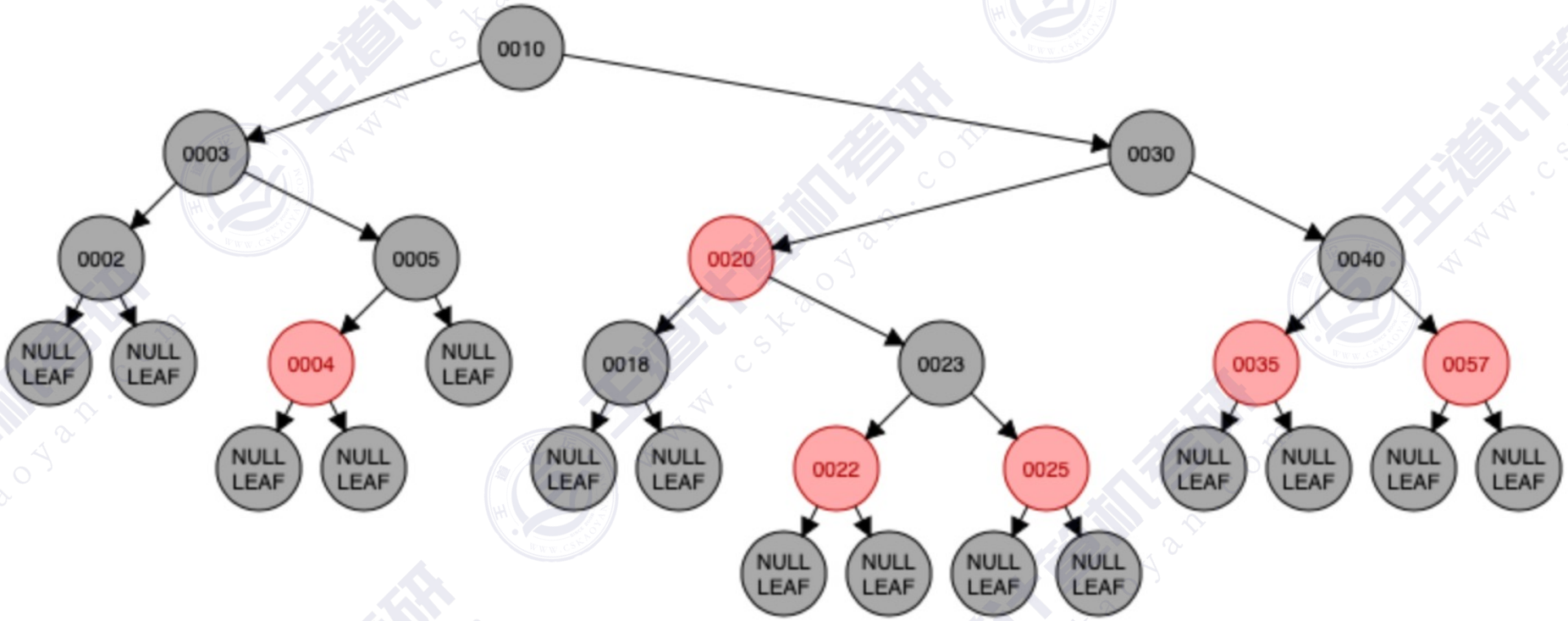
②右单旋的结果：儿子辈分再升高

- 黑叔：旋转+染色
  - LR型：左、右双旋，儿换爷+染色

红黑树的插入

从一棵空的红黑树开始，插入：20, 10, 5, 30, 40, 57, 3, 2, 4, 35, 25, 18, 22, 23, 24, 19, 18

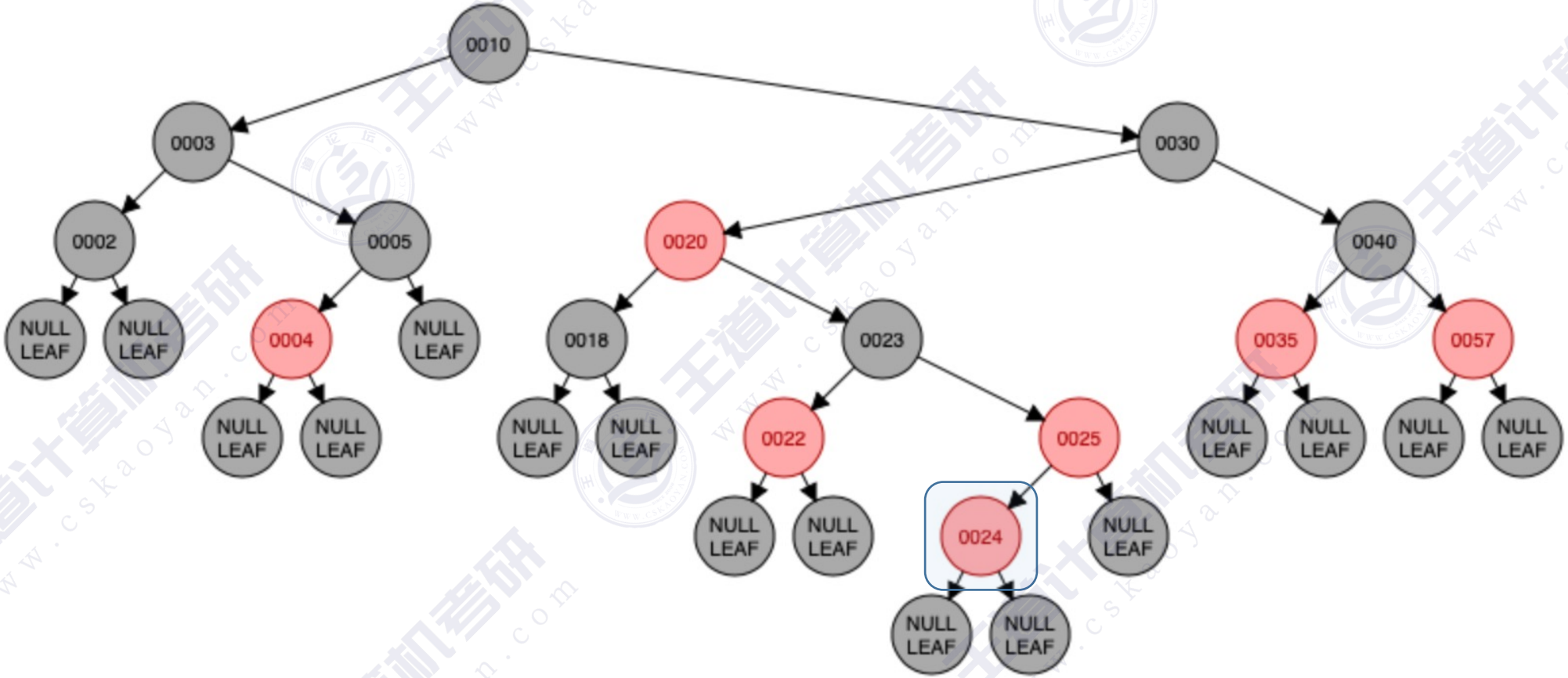
原来的儿和爷“染色”



- 黑叔：旋转+染色
  - LR型：左、右双旋，儿换爷+染色

红黑树的插入

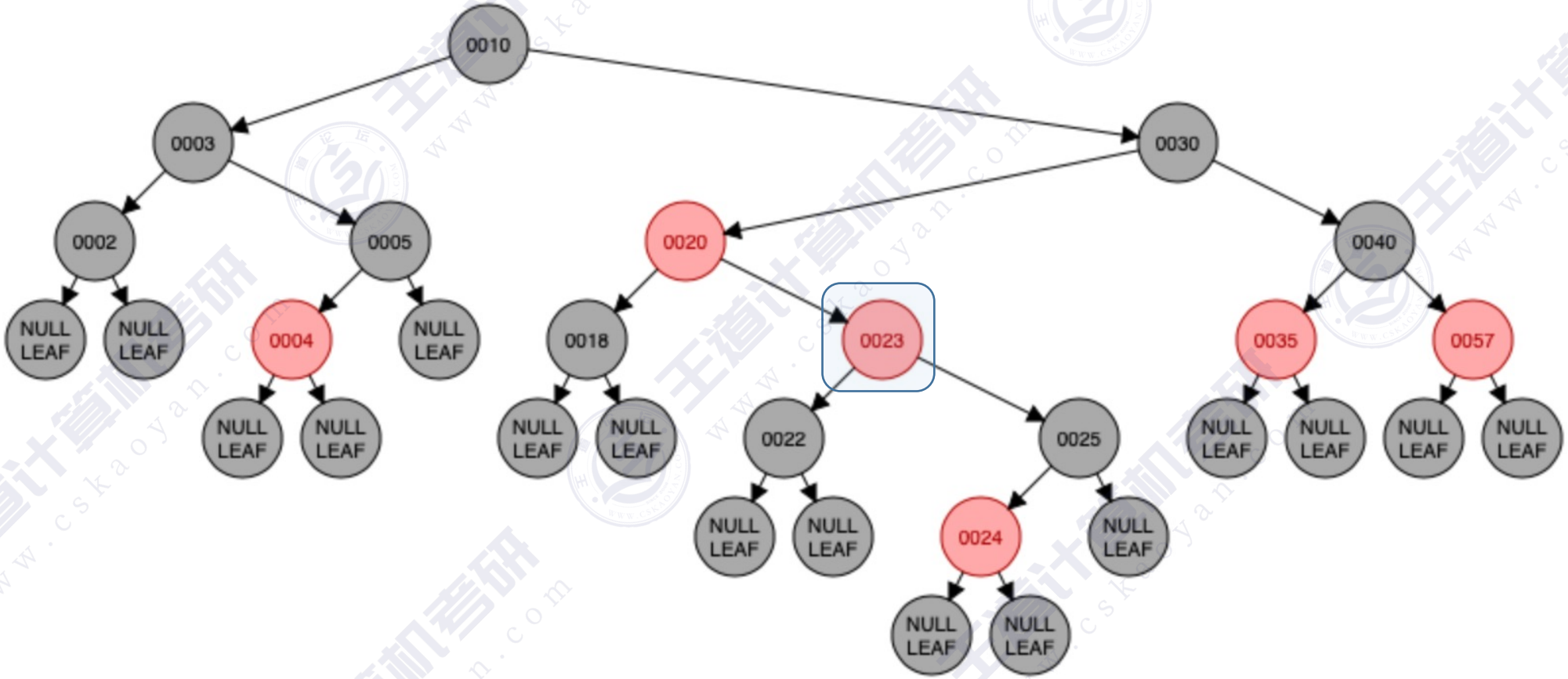
从一棵空的红黑树开始，插入：20, 10, 5, 30, 40, 57, 3, 2, 4, 35, 25, 18, 22, 23, 24, 19, 18



- 红叔：染色+变新
  - 叔父爷染色，爷变为新结点

红黑树的插入

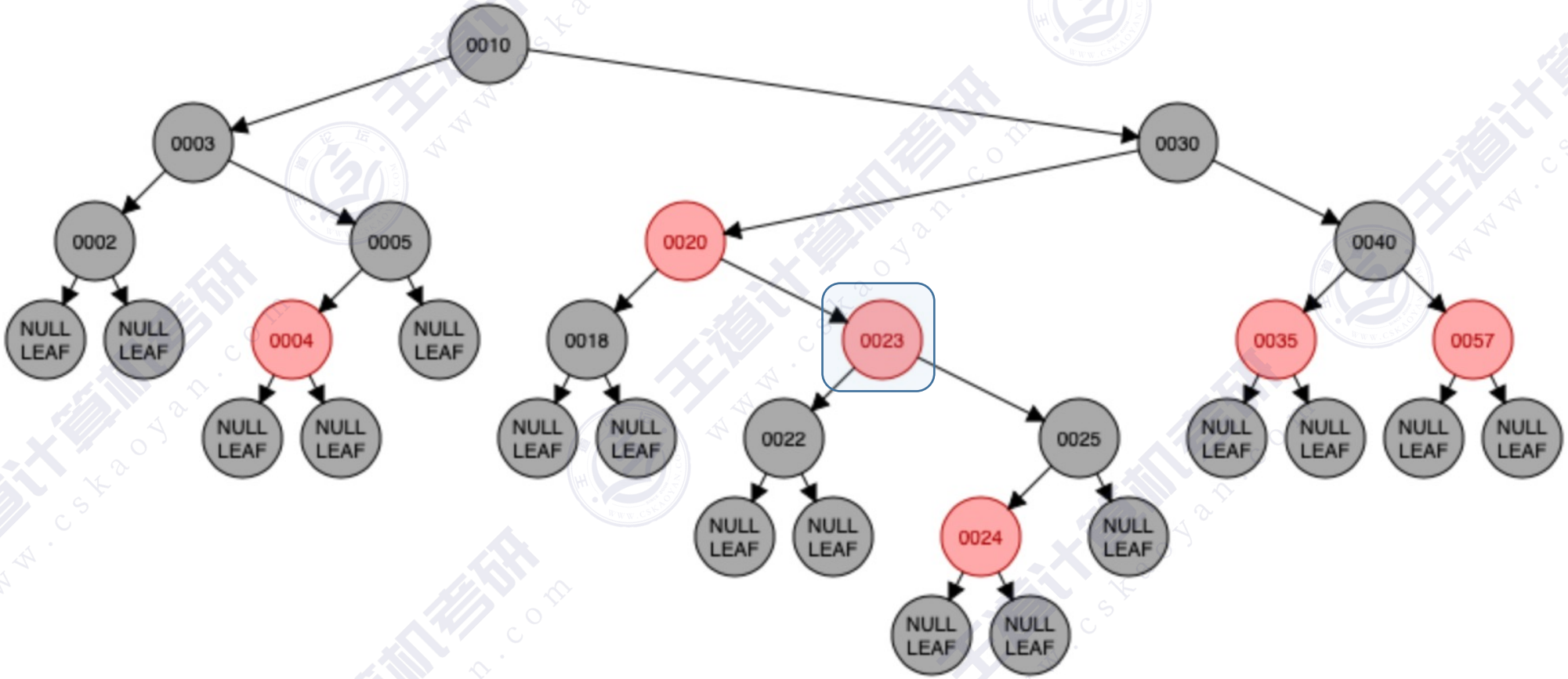
从一棵空的红黑树开始，插入：20, 10, 5, 30, 40, 57, 3, 2, 4, 35, 25, 18, 22, 23, 24, 19, 18



- 红叔：染色+变新
  - 叔父爷染色，爷变为新结点

红黑树的插入

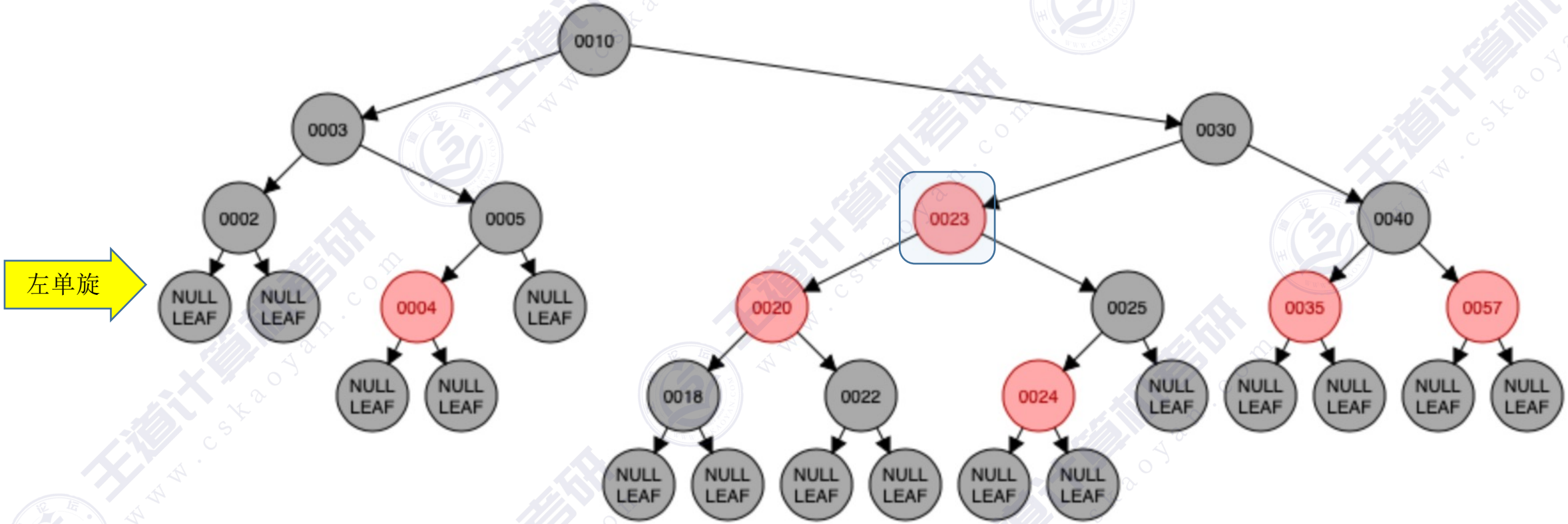
从一棵空的红黑树开始，插入：20, 10, 5, 30, 40, 57, 3, 2, 4, 35, 25, 18, 22, 23, 24, 19, 18



- 黑叔：旋转+染色
  - LR型：左、右双旋，儿换爷+染色

红黑树的插入

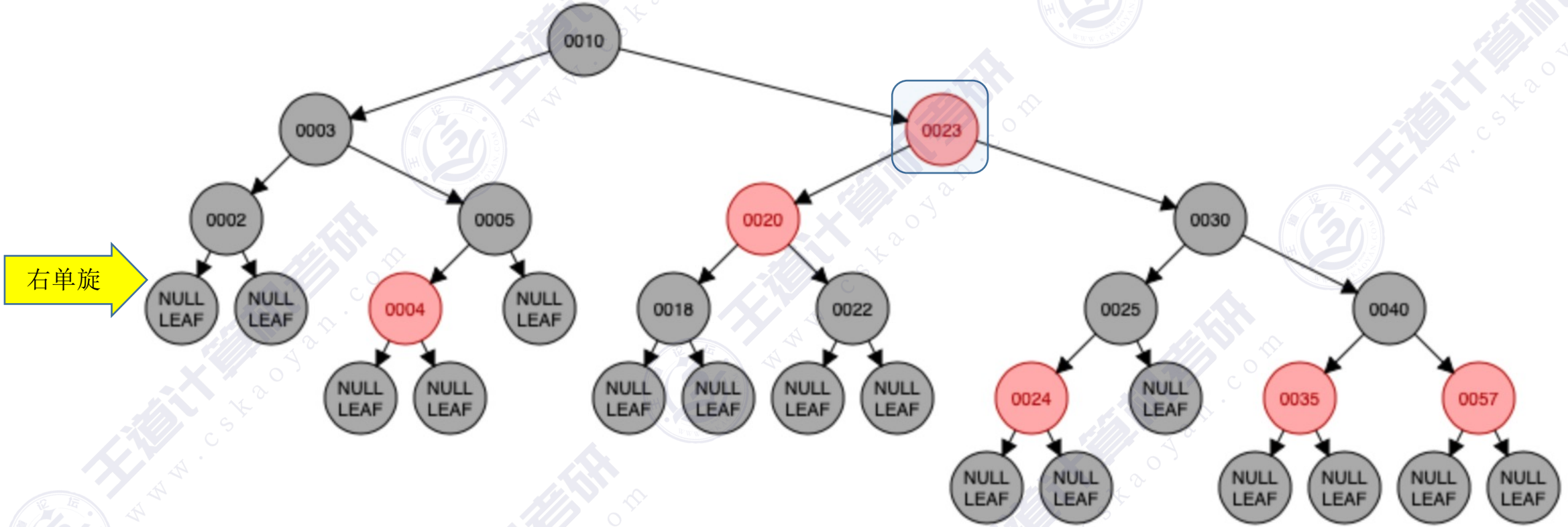
从一棵空的红黑树开始，插入：20, 10, 5, 30, 40, 57, 3, 2, 4, 35, 25, 18, 22, 23, 24, 19, 18



- 黑叔：旋转+染色
  - LR型：左、右双旋，儿换爷+染色

红黑树的插入

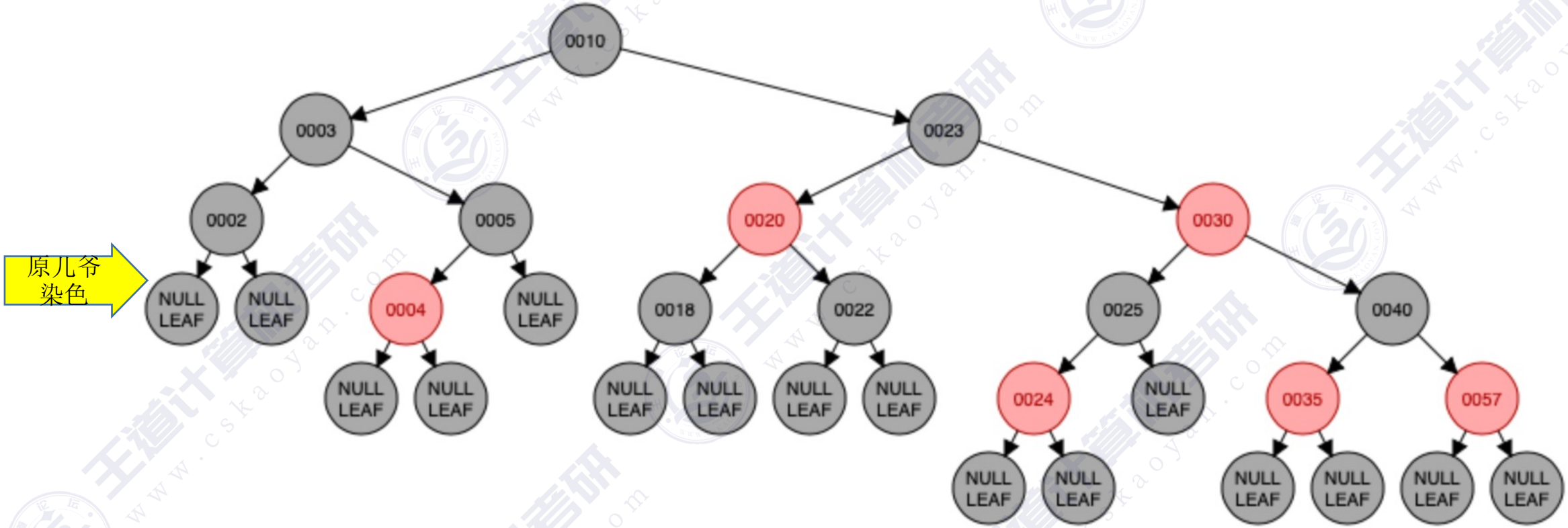
从一棵空的红黑树开始，插入：20, 10, 5, 30, 40, 57, 3, 2, 4, 35, 25, 18, 22, 23, 24, 19, 18



- 黑叔：旋转+染色
  - LR型：左、右双旋，儿换爷+染色

红黑树的插入

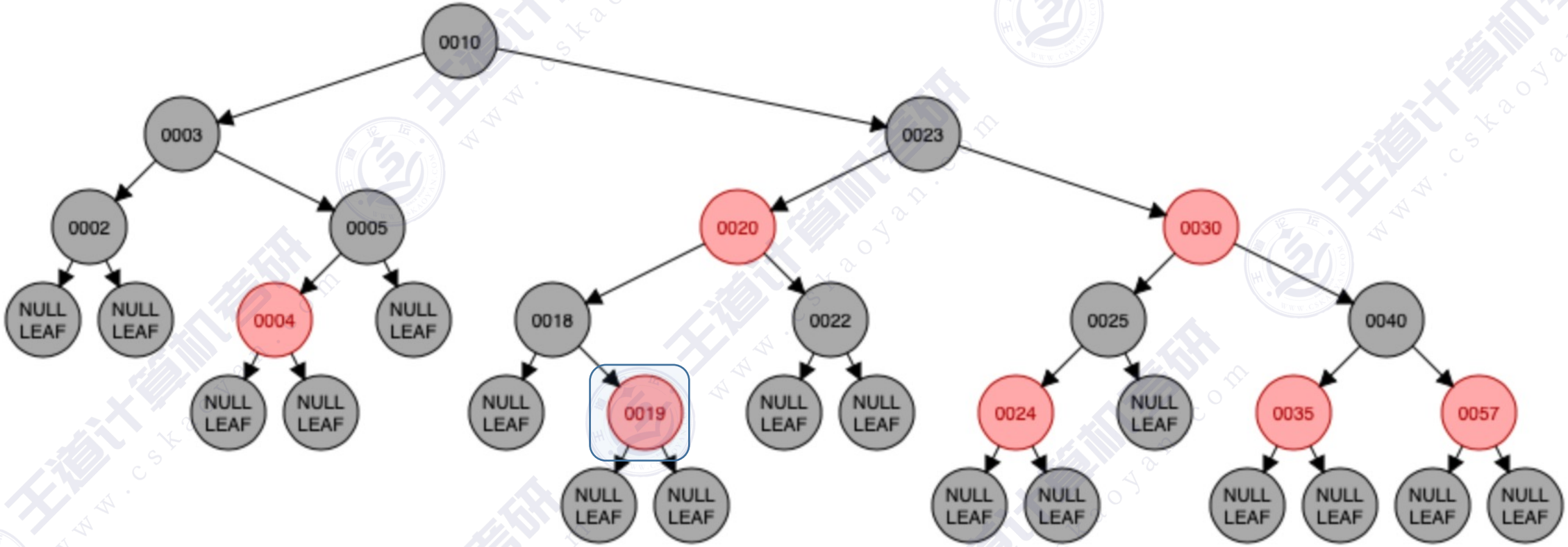
从一棵空的红黑树开始，插入：20, 10, 5, 30, 40, 57, 3, 2, 4, 35, 25, 18, 22, 23, 24, 19, 18



- 黑叔：旋转+染色
  - LR型：左、右双旋，儿换爷+染色

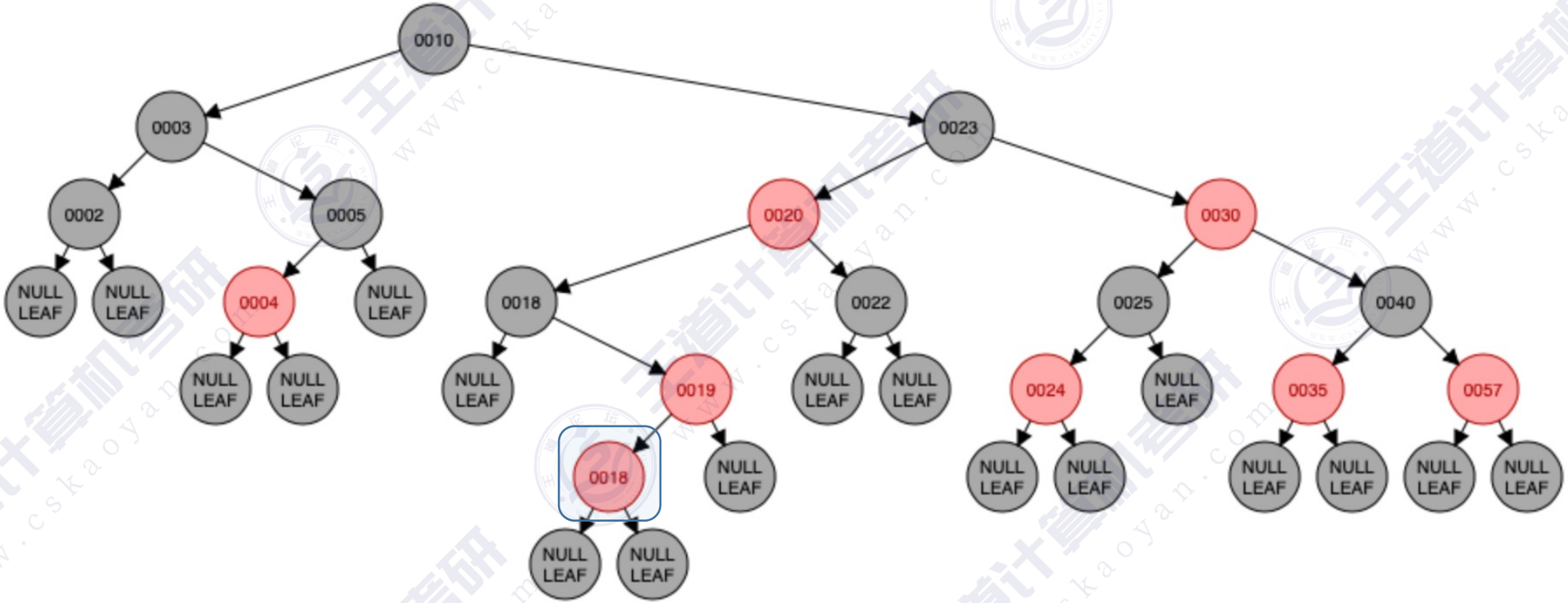
红黑树的插入

从一棵空的红黑树开始，插入：20, 10, 5, 30, 40, 57, 3, 2, 4, 35, 25, 18, 22, 23, 24, 19, 18



红黑树的插入

从一棵空的红黑树开始，插入：20, 10, 5, 30, 40, 57, 3, 2, 4, 35, 25, 18, 22, 23, 24, 19, 18

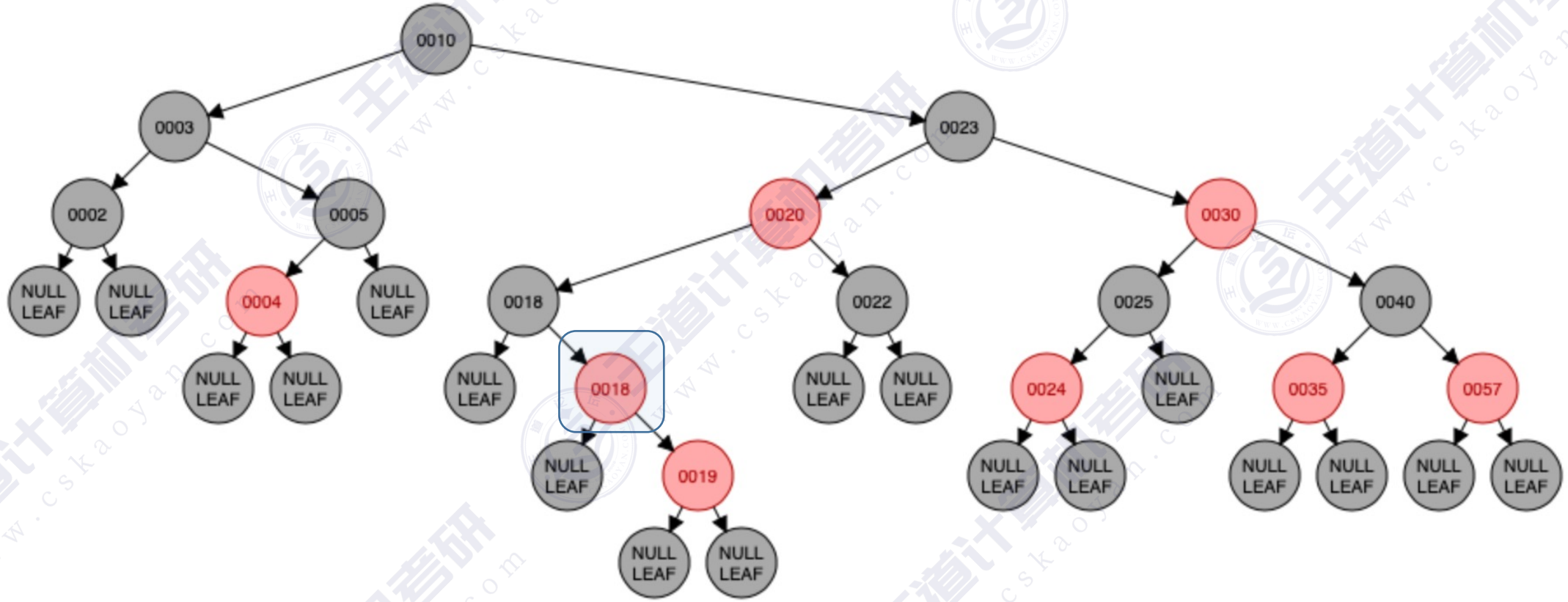


- 黑叔：旋转+染色
  - RL型：右、左双旋，儿换爷+染色

红黑树的插入

从一棵空的红黑树开始，插入：20, 10, 5, 30, 40, 57, 3, 2, 4, 35, 25, 18, 22, 23, 24, 19, 18

右单旋

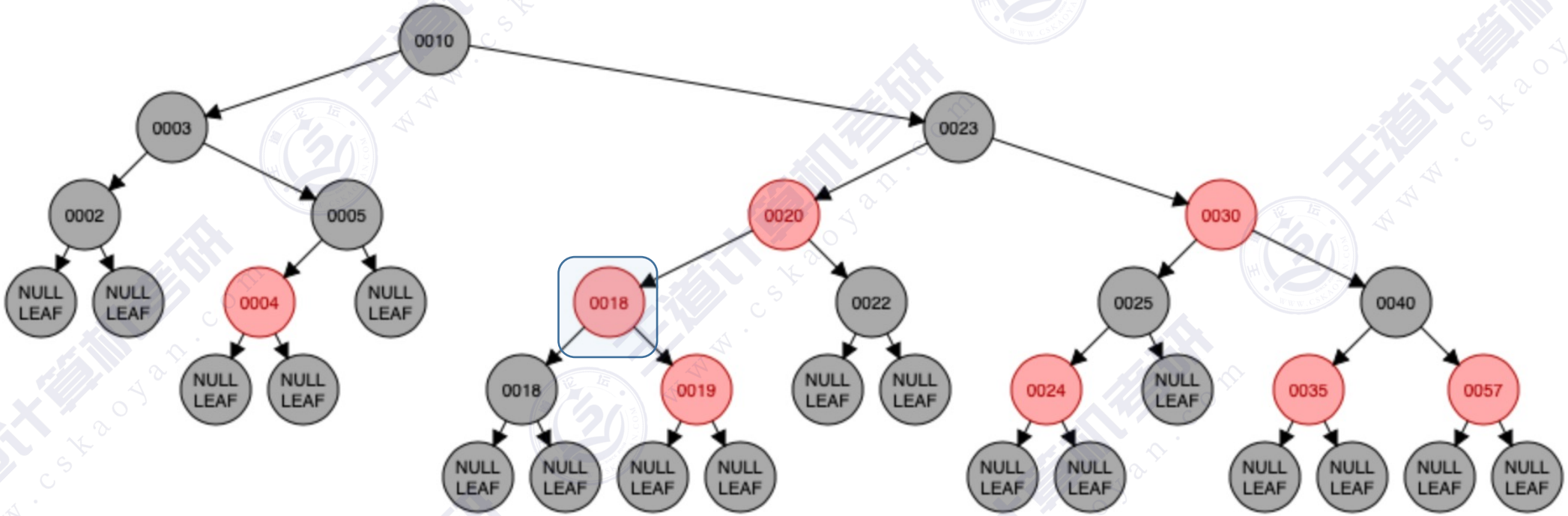


- 黑叔：旋转+染色
  - RL型：右、左双旋，儿换爷+染色

红黑树的插入

从一棵空的红黑树开始，插入：20, 10, 5, 30, 40, 57, 3, 2, 4, 35, 25, 18, 22, 23, 24, 19, 18

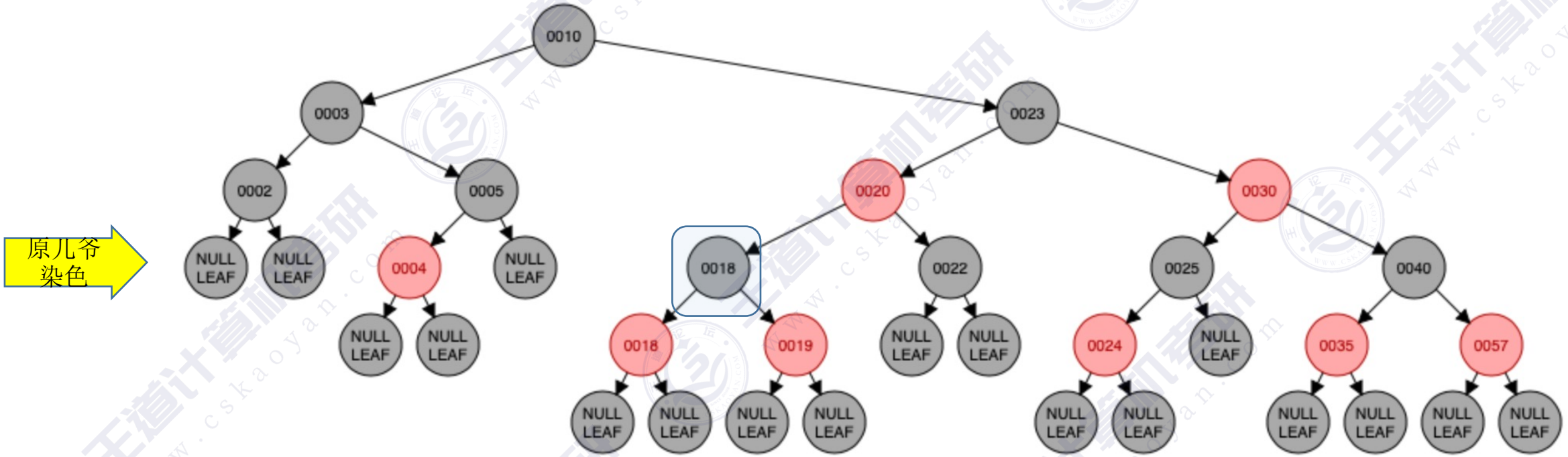
左单旋



- 黑叔：旋转+染色
  - RL型：右、左双旋，儿换爷+染色

红黑树的插入

从一棵空的红黑树开始，插入：20, 10, 5, 30, 40, 57, 3, 2, 4, 35, 25, 18, 22, 23, 24, 19, 18



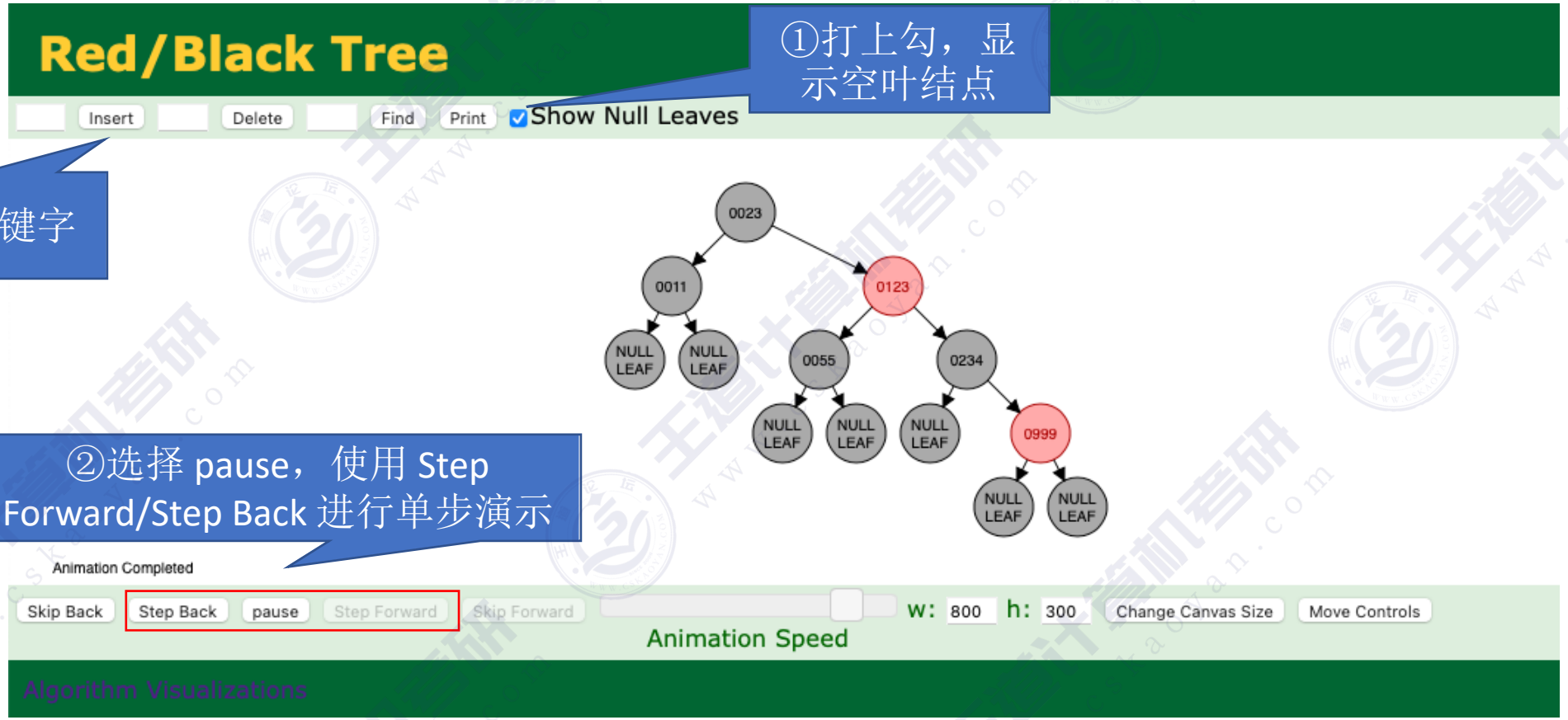
- 黑叔：旋转+染色
  - RL型：右、左双旋，儿换爷+染色

知识回顾与重要考点


红黑树RBT
相关概念
黑高 —— 一个结点的“黑高”：从某结点出发（不含该结点）到达任一空叶结点的路径上黑结点总数
叶结点 —— 在RBT中，“叶结点”通常指“失败结点”（又名：空叶结点/NULL结点/外部结点）
定义
左根右 —— RBT是一种BST，需满足 左≤根≤右
根叶黑 —— 根节点、叶结点一定是黑色
不红红 —— 任何一条查找路径上不能连续出现两个红结点
黑路同 —— 从任一结点出发，到达任一空叶结点的路径上经过的黑结点数量都相同
性质
从根节点到叶结点的最长路径不大于最短路径的2倍
有n个内部节点的红黑树高度 h≤2log (n+1)
插入
新结点为根染黑
新结点非根染红，若插入后违反“不红红”，则看叔叔的脸色
黑叔：旋转+染色
LL —— 右单旋，父换爷+染色
RR —— 左单旋，父换爷+染色
LR —— 左、右双旋，儿换爷+染色
RL —— 右、左双旋，儿换爷+染色
红叔：染色+变新 —— 叔父爷染色，爷变为新结点

红黑树练习方法（插入操作）

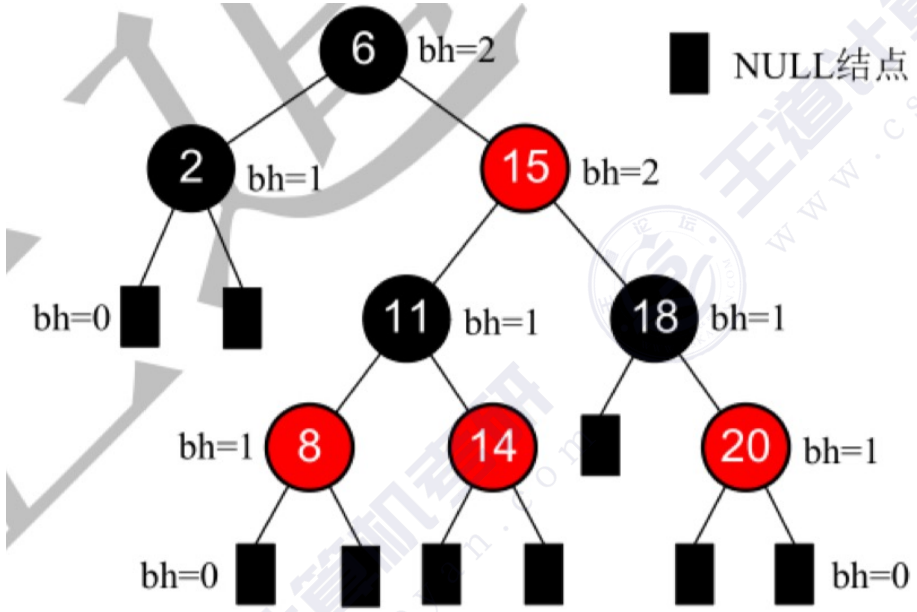
课件中的例子，插入：20, 10, 5, 30, 40, 57, 3, 2, 4, 35, 25, 18, 22, 23, 24, 19, 18

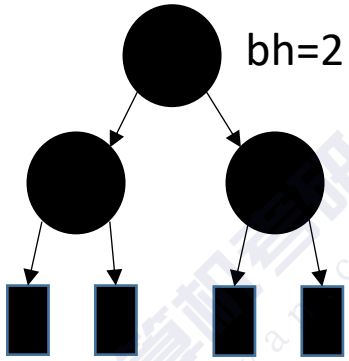


自我训练方法：你可以自己设计一些插入序列，每一次插入后，都先思考应该如何调整，然后在网站中演示验证。插入新元素时，尽可能覆盖 黑叔LL/RR/LR/RL、红叔 的情况。

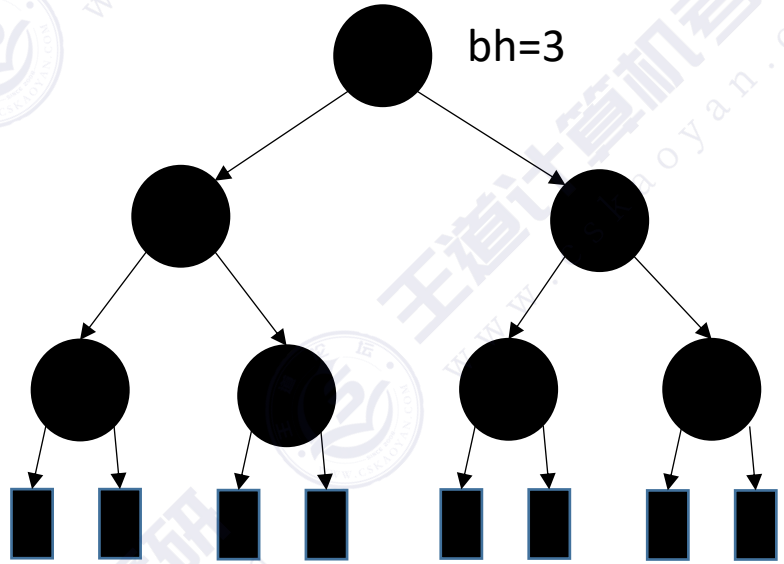
https://www.cs.usfca.edu/~galles/visualization/RedBlack.html

与“黑高”相关的推论





根节点黑高=2，内部结点数最少的情况



根节点黑高=3，内部结点数最少的情况

结点的黑高 bh —— 从某结点出发（不含该结点）到达任一叶结点的路径上黑结点总数

思考：根节点黑高为 h 的红黑树，内部结点数（关键字）至少有多少个？
回答：内部结点数最少的情况——总共h层黑结点的满树形态
结论：若根节点黑高为h，内部结点数（关键字）最少有 2^h-1 个

红黑树的定义→性质

红黑树是二叉排序树 → 左子树结点值 ≤ 根结点值 ≤ 右子树结点值

与普通BST相比，有什么要求 →

①每个结点或是红色，或是黑色的
②根节点是黑色的
③叶结点（外部结点、NULL结点、失败结点）均是黑色的
④不存在两个相邻的红结点（即红结点的父节点和孩子结点均是黑色）
⑤对每个结点，从该节点到任一叶结点的简单路径上，所含黑结点的数目相同

左根右，根叶黑
不红红，黑路同

张口就是freestyle

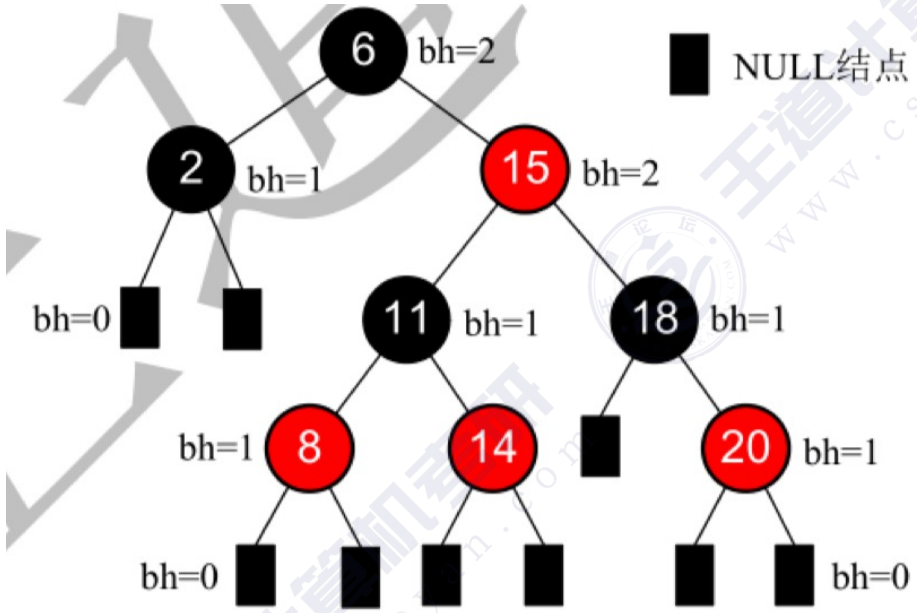
→ 性质1：从根节点到叶结点的最长路径不大于最短路径的2倍
性质2：有n个内部节点的红黑树高度 $h \leq 2log_2(n+1)$

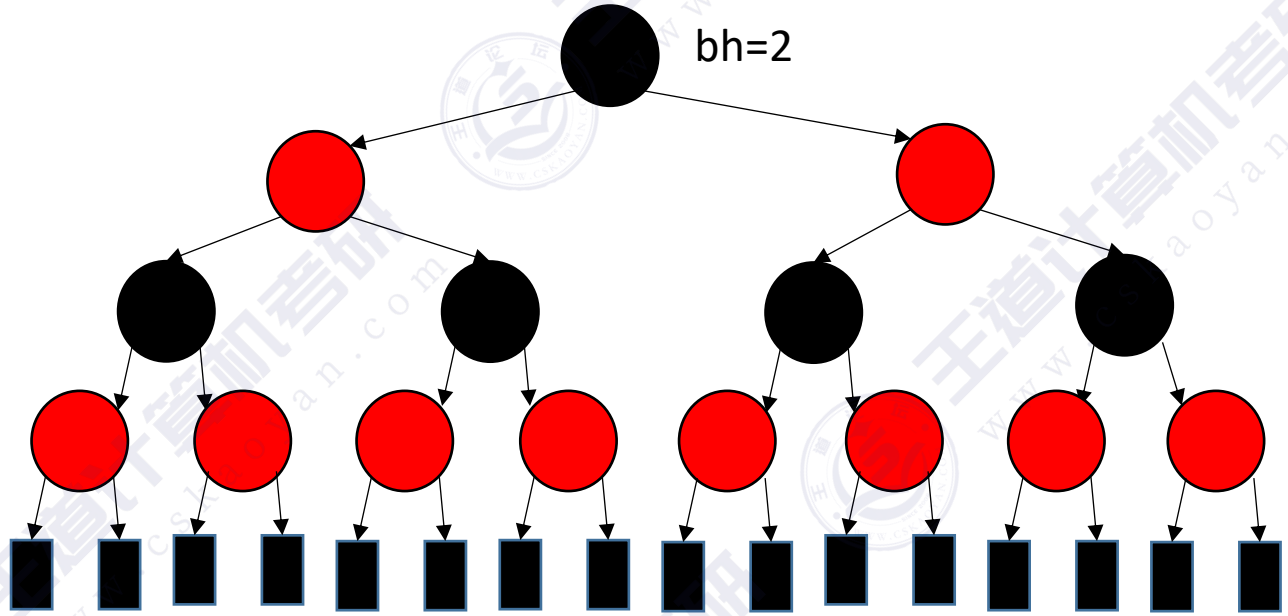
→ 红黑树查找操作时间复杂度 = $O(log_2 n)$

性质1证明：任何一条查找失败路径上黑结点数量都相同，而路径上不能连续出现两个红结点，即红结点只能穿插在各个黑结点中间

性质2证明：若红黑树总高度=h，则根节点黑高 ≥ h/2，因此内部结点数 $n \geq 2^{h/2}-1$，由此推出 $h \leq 2log_2(n+1)$

与“黑高”相关的推论





根节点黑高=2，内部结点数最多的情况

结点的黑高 bh —— 从某结点出发（不含该结点）到达任一叶结点的路径上黑结点总数

思考：根节点黑高为 h 的红黑树，内部结点数（关键字）至多有多少个？
回答：内部结点数最多的情况——h层黑结点，每一层黑结点下面都铺满一层红结点。共2h层的满树形态
结论：若根节点黑高为h，内部结点数（关键字）最多有 $2^{2h}-1$ 个