

本节内容

## 平衡二叉树

### 删除操作

王道考研/CSKAOYAN.COM

1

## 平衡二叉树的插入&删除

平衡二叉树的**插入**操作:

- 插入新结点后,要保持**二叉排序树**的特性不变(左<中<右)
- 若插入新结点导致**不平衡**,则需要**调整平衡**

平衡二叉树的**删除**操作:

- 删除结点后,要保持**二叉排序树**的特性不变(左<中<右)
- 若删除结点导致**不平衡**,则需要**调整平衡**

不一样?



王道考研/CSKAOYAN.COM

2

## 平衡二叉树的删除

平衡二叉树的删除操作：

- 删除结点后，要保持二叉排序树的特性不变（左<中<右）
- 若删除结点导致不平衡，则需要调整平衡

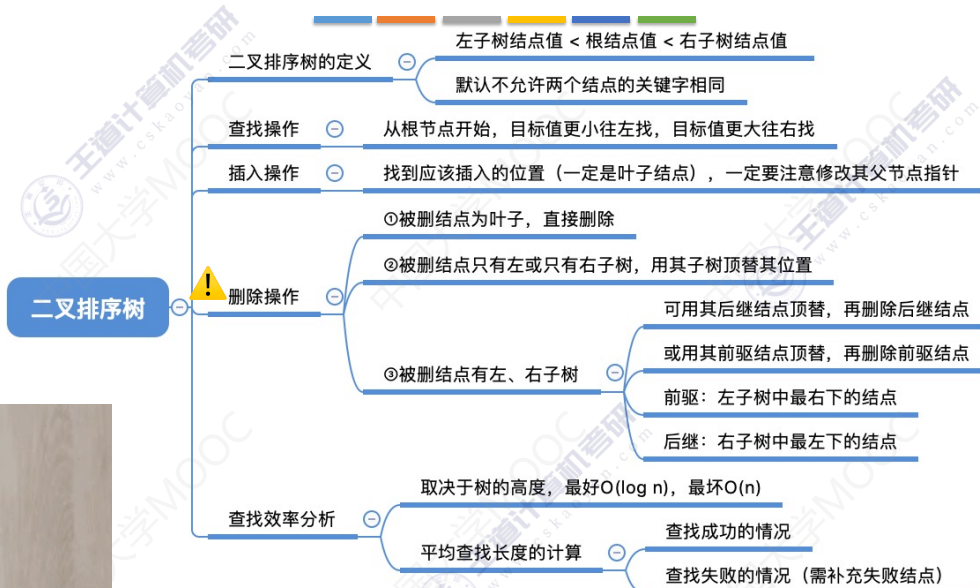
平衡二叉树的删除操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
- ②一路向北找到最小不平衡子树，找不到就完结撒花
- ③找最小不平衡子树下，“个头”最高的儿子、孙子
- ④根据孙子的位置，调整平衡（LL/RR/LR/RL）
- ⑤如果不平衡向上传导，继续②

王道考研/CSKAOYAN.COM

3

## 暂停偷看：二叉排序树的删除操作



王道考研/CSKAOYAN.COM

4

## AVL树删除操作——例1

平衡二叉树的删除操作具体步骤：

### ①删除结点（方法同“二叉排序树”）

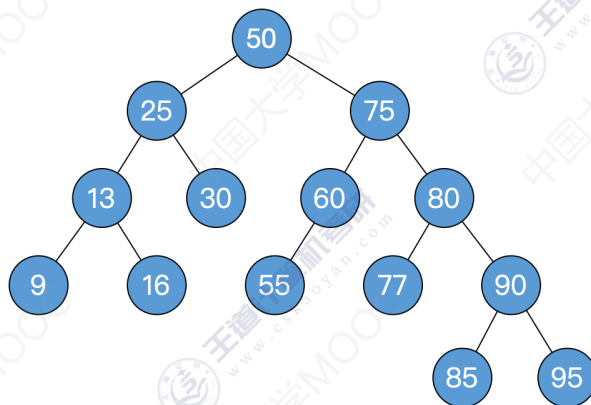
- 若删除的结点是叶子，直接删。
- 若删除的结点只有一个子树，用子树顶替删除位置
- 若删除的结点有两棵子树，用前驱（或后继）结点顶替，并转换为对前驱（或后继）结点的删除。

②一路向北找到最小不平衡子树，找不到就完结撒花

③找最小不平衡子树下，“个头”最高的儿子、孙子

④根据孙子的位置，调整平衡（LL/RR/LR/RL）

⑤如果不平衡向上传导，继续②



王道考研/CSKAOYAN.COM

5

## AVL树删除操作——例1

平衡二叉树的删除操作具体步骤：

### ①删除结点（方法同“二叉排序树”）

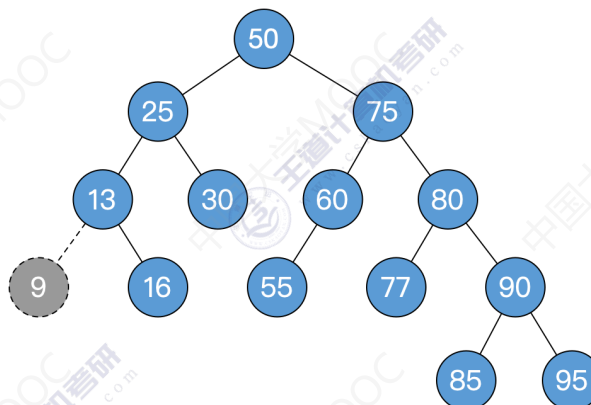
- 若删除的结点是叶子，直接删。
- 若删除的结点只有一个子树，用子树顶替删除位置
- 若删除的结点有两棵子树，用前驱（或后继）结点顶替，并转换为对前驱（或后继）结点的删除。

②一路向北找到最小不平衡子树，找不到就完结撒花

③找最小不平衡子树下，“个头”最高的儿子、孙子

④根据孙子的位置，调整平衡（LL/RR/LR/RL）

⑤如果不平衡向上传导，继续②



王道考研/CSKAOYAN.COM

6

## AVL树删除操作——例1

平衡二叉树的删除操作具体步骤：

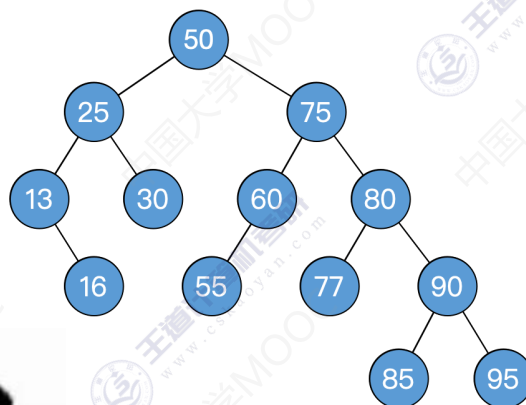
①删除结点（方法同“二叉排序树”）

②一路向北找到最小不平衡子树，找不到就完结撒花

③找最小不平衡子树下，“个头”最高的儿子、孙子

④根据孙子的位置，调整平衡（LL/RR/LR/RL）

⑤如果不平衡向上传导，继续②



王道考研/CSKAOYAN.COM

7

## AVL树删除操作——例2

平衡二叉树的删除操作具体步骤：

①删除结点（方法同“二叉排序树”）

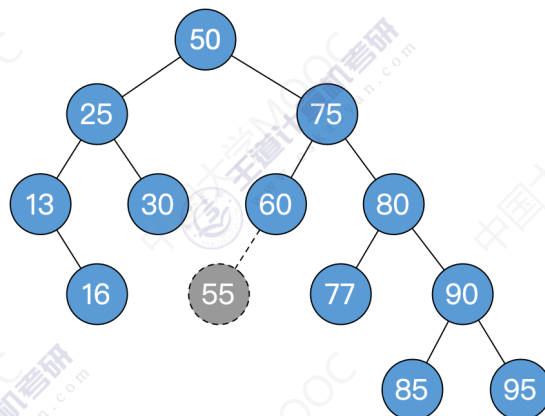
- 若删除的结点是叶子，直接删。
- 若删除的结点只有一个子树，用子树顶替删除位置
- 若删除的结点有两棵子树，用前驱（或后继）结点顶替，并转换为对前驱（或后继）结点的删除。

②一路向北找到最小不平衡子树，找不到就完结撒花

③找最小不平衡子树下，“个头”最高的儿子、孙子

④根据孙子的位置，调整平衡（LL/RR/LR/RL）

⑤如果不平衡向上传导，继续②



王道考研/CSKAOYAN.COM

8

## AVL树删除操作——例2

平衡二叉树的删除操作具体步骤：

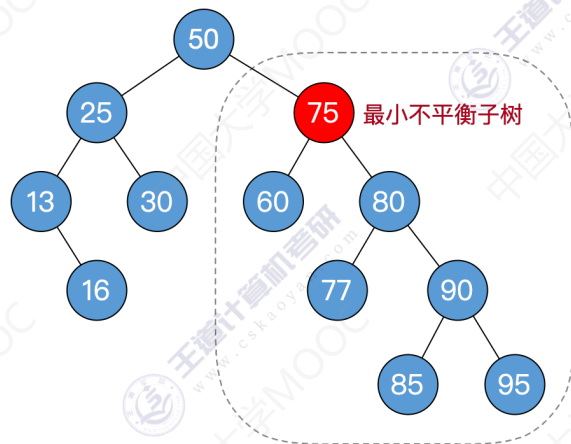
①删除结点（方法同“二叉排序树”）

②一路向北找到最小不平衡子树，找不到就完结撒花

③找最小不平衡子树下，“个头”最高的儿子、孙子

④根据孙子的位置，调整平衡（LL/RR/LR/RL）

⑤如果不平衡向上传导，继续②



王道考研/CSKAOYAN.COM

9

## AVL树删除操作——例2

平衡二叉树的删除操作具体步骤：

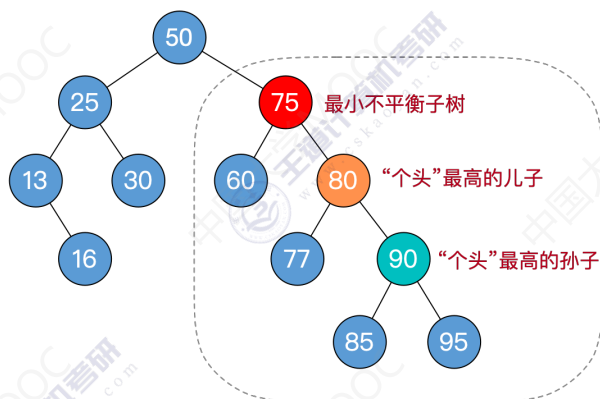
①删除结点（方法同“二叉排序树”）

②一路向北找到最小不平衡子树，找不到就完结撒花

③找最小不平衡子树下，“个头”最高的儿子、孙子

④根据孙子的位置，调整平衡（LL/RR/LR/RL）

⑤如果不平衡向上传导，继续②



王道考研/CSKAOYAN.COM

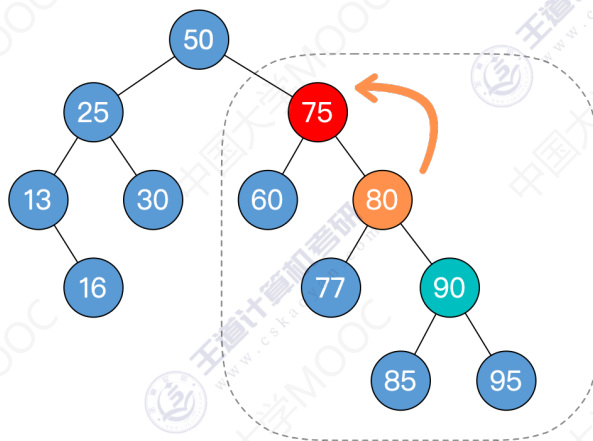
10



## AVL树删除操作——例2

平衡二叉树的删除操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
- ②一路向北找到最小不平衡子树，找不到就完结撒花
- ③找最小不平衡子树下，“个头”最高的儿子、孙子
- ④根据孙子的位置，调整平衡（LL/RR/LR/RL）
  - 孙子在LL：儿子右单旋
  - 孙子在RR：儿子左单旋
  - 孙子在LR：孙子先左旋，再右旋
  - 孙子在RL：孙子先右旋，再左旋
- ⑤如果不平衡向上传导，继续②



孙子在RR，儿子左单旋

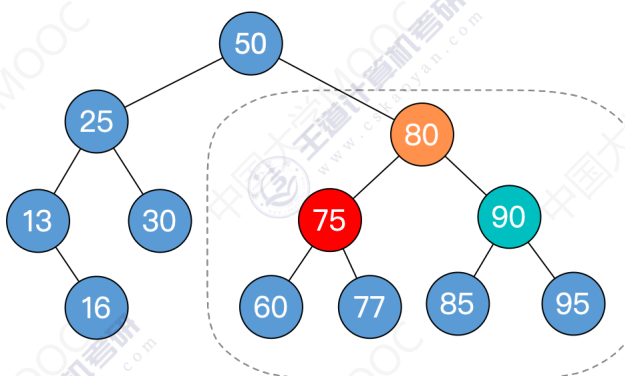
王道考研/CSKAOYAN.COM

11

## AVL树删除操作——例2

平衡二叉树的删除操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
- ②一路向北找到最小不平衡子树，找不到就完结撒花
- ③找最小不平衡子树下，“个头”最高的儿子、孙子
- ④根据孙子的位置，调整平衡（LL/RR/LR/RL）
  - 孙子在LL：儿子右单旋
  - 孙子在RR：儿子左单旋
  - 孙子在LR：孙子先左旋，再右旋
  - 孙子在RL：孙子先右旋，再左旋
- ⑤如果不平衡向上传导，继续②



最小不平衡子树恢复平衡

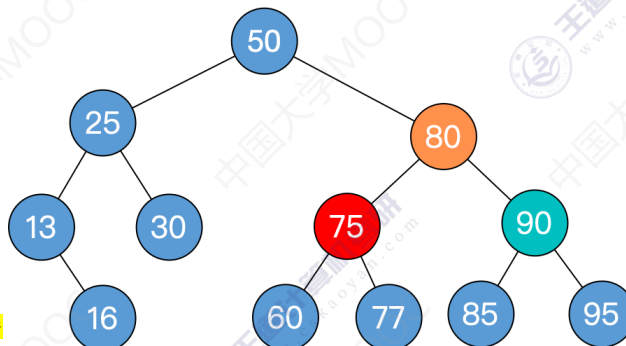
王道考研/CSKAOYAN.COM

12

## AVL树删除操作——例2

平衡二叉树的删除操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
  - ②一路向北找到最小不平衡子树，找不到就完结撒花
  - ③找最小不平衡子树下，“个头”最高的儿子、孙子
  - ④根据孙子的位置，调整平衡（LL/RR/LR/RL）
  - ⑤如果不平衡向上传导，继续②
- 对最小不平衡子树的旋转可能导致树变矮，从而导致上层祖先不平衡（不平衡向上传递）



Over !



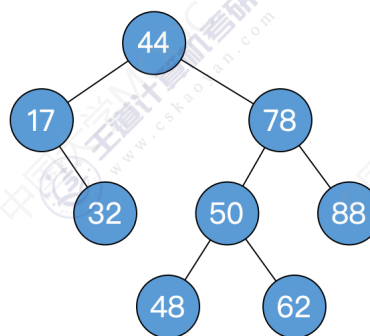
王道考研/CSKAOYAN.COM

13

## AVL树删除操作——例3

平衡二叉树的删除操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
- 若删除的结点是叶子，直接删。
  - 若删除的结点只有一个子树，用子树顶替删除位置
  - 若删除的结点有两棵子树，用前驱（或后继）结点顶替，并转换为对前驱（或后继）结点的删除。
- ②一路向北找到最小不平衡子树，找不到就完结撒花
  - ③找最小不平衡子树下，“个头”最高的儿子、孙子
  - ④根据孙子的位置，调整平衡（LL/RR/LR/RL）
  - ⑤如果不平衡向上传导，继续②



王道考研/CSKAOYAN.COM

14

### AVL树删除操作——例3

平衡二叉树的删除操作具体步骤：

①删除结点（方法同“二叉排序树”）

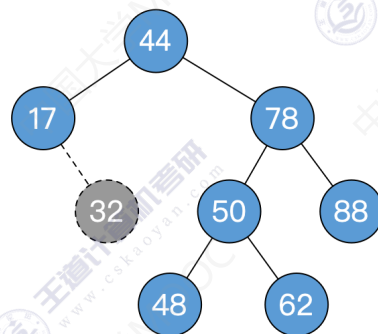
- 若删除的结点是叶子，直接删。
- 若删除的结点只有一个子树，用子树顶替删除位置
- 若删除的结点有两棵子树，用前驱（或后继）结点顶替，并转换为对前驱（或后继）结点的删除。

②一路向北找到最小不平衡子树，找不到就完结撒花

③找最小不平衡子树下，“个头”最高的儿子、孙子

④根据孙子的位置，调整平衡（LL/RR/LR/RL）

⑤如果不平衡向上传导，继续②



王道考研/CSKAOYAN.COM

15

### AVL树删除操作——例3

平衡二叉树的删除操作具体步骤：

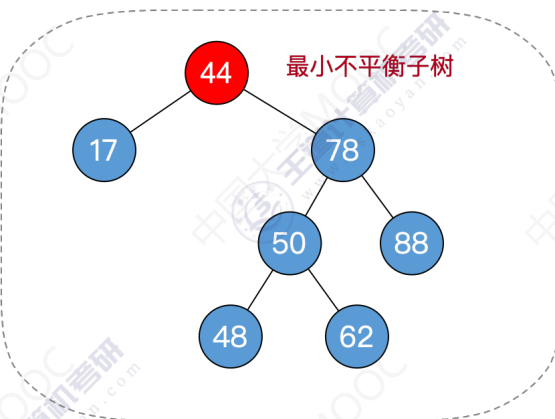
①删除结点（方法同“二叉排序树”）

②一路向北找到最小不平衡子树，找不到就完结撒花

③找最小不平衡子树下，“个头”最高的儿子、孙子

④根据孙子的位置，调整平衡（LL/RR/LR/RL）

⑤如果不平衡向上传导，继续②



王道考研/CSKAOYAN.COM

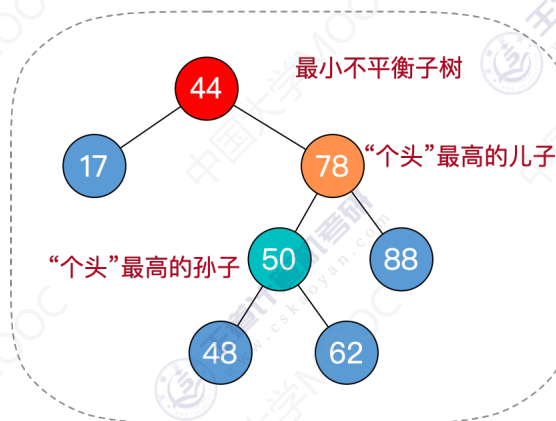
16



### AVL树删除操作——例3

平衡二叉树的删除操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
- ②一路向北找到最小不平衡子树，找不到就完结撒花
- ③找最小不平衡子树下，“个头”最高的儿子、孙子
- ④根据孙子的位置，调整平衡（LL/RR/LR/RL）
- ⑤如果不平衡向上传导，继续②



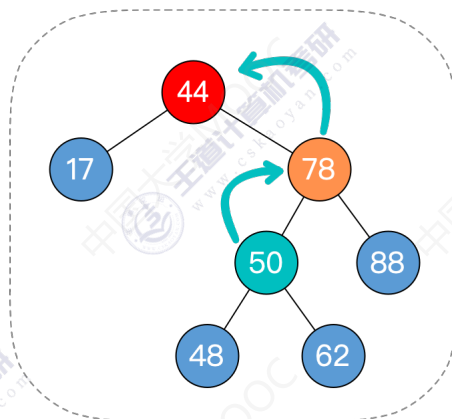
王道考研/CSKAOYAN.COM

17

### AVL树删除操作——例3

平衡二叉树的删除操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
- ②一路向北找到最小不平衡子树，找不到就完结撒花
- ③找最小不平衡子树下，“个头”最高的儿子、孙子
- ④根据孙子的位置，调整平衡（LL/RR/LR/RL）
  - 孙子在LL：儿子右单旋
  - 孙子在RR：儿子左单旋
  - 孙子在LR：孙子先左旋，再右旋
  - 孙子在RL：孙子先右旋，再左旋
- ⑤如果不平衡向上传导，继续②



孙子在RL，孙子先右旋，再左旋

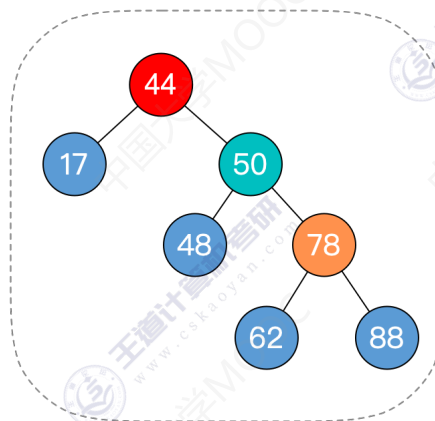
王道考研/CSKAOYAN.COM

18

### AVL树删除操作——例3

平衡二叉树的删除操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
- ②一路向北找到最小不平衡子树，找不到就完结撒花
- ③找最小不平衡子树下，“个头”最高的儿子、孙子
- ④根据孙子的位置，调整平衡（LL/RR/LR/RL）
  - 孙子在LL：儿子右单旋
  - 孙子在RR：儿子左单旋
  - 孙子在LR：孙子先左旋，再右旋
  - 孙子在RL：孙子先右旋，再左旋
- ⑤如果不平衡向上传导，继续②



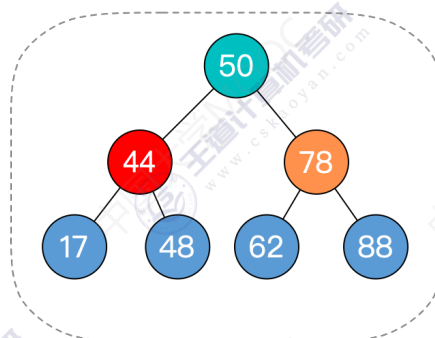
王道考研/CSKAOYAN.COM

19

### AVL树删除操作——例3

平衡二叉树的删除操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
- ②一路向北找到最小不平衡子树，找不到就完结撒花
- ③找最小不平衡子树下，“个头”最高的儿子、孙子
- ④根据孙子的位置，调整平衡（LL/RR/LR/RL）
  - 孙子在LL：儿子右单旋
  - 孙子在RR：儿子左单旋
  - 孙子在LR：孙子先左旋，再右旋
  - 孙子在RL：孙子先右旋，再左旋
- ⑤如果不平衡向上传导，继续②



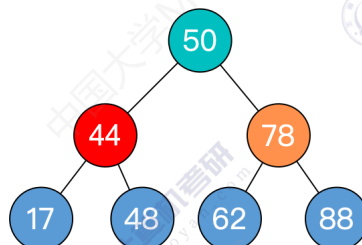
王道考研/CSKAOYAN.COM

20

### AVL树删除操作——例3

平衡二叉树的删除操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
  - ②一路向北找到最小不平衡子树，找不到就完结撒花
  - ③找最小不平衡子树下，“个头”最高的儿子、孙子
  - ④根据孙子的位置，调整平衡（LL/RR/LR/RL）
  - ⑤如果不平衡向上传导，继续②
- 对最小不平衡子树的旋转可能导致树变矮，从而导致上层祖先不平衡（不平衡向上传递）



Over !



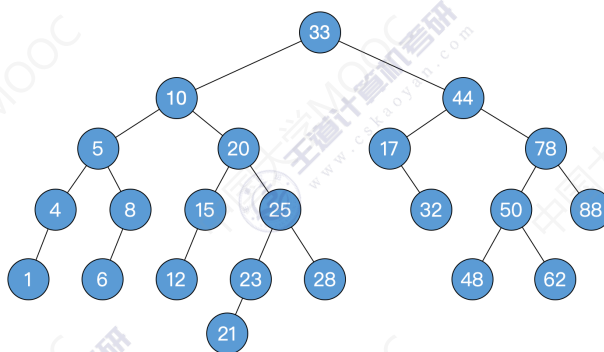
王道考研/CSKAOYAN.COM

21

### AVL树删除操作——例4

平衡二叉树的删除操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
- 若删除的结点是叶子，直接删。
  - 若删除的结点只有一个子树，用子树顶替删除位置
  - 若删除的结点有两棵子树，用前驱（或后继）结点顶替，并转换为对前驱（或后继）结点的删除。
- ②一路向北找到最小不平衡子树，找不到就完结撒花
  - ③找最小不平衡子树下，“个头”最高的儿子、孙子
  - ④根据孙子的位置，调整平衡（LL/RR/LR/RL）
  - ⑤如果不平衡向上传导，继续②



王道考研/CSKAOYAN.COM

22

## AVL树删除操作——例4

平衡二叉树的删除操作具体步骤：

### ①删除结点（方法同“二叉排序树”）

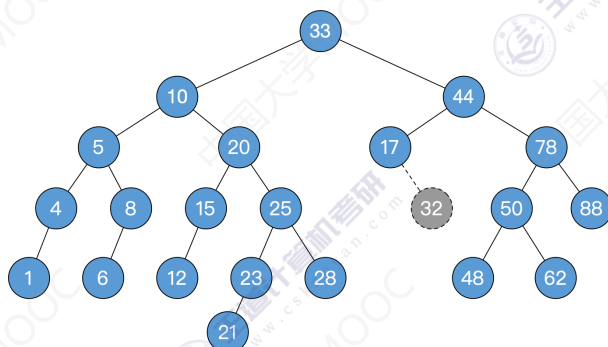
- 若删除的结点是叶子，直接删。
- 若删除的结点只有一个子树，用子树顶替删除位置
- 若删除的结点有两棵子树，用前驱（或后继）结点顶替，并转换为对前驱（或后继）结点的删除。

②一路向北找到最小不平衡子树，找不到就完结撒花

③找最小不平衡子树下，“个头”最高的儿子、孙子

④根据孙子的位置，调整平衡（LL/RR/LR/RL）

⑤如果不平衡向上传导，继续②



王道考研/CSKAOYAN.COM

23

## AVL树删除操作——例4

平衡二叉树的删除操作具体步骤：

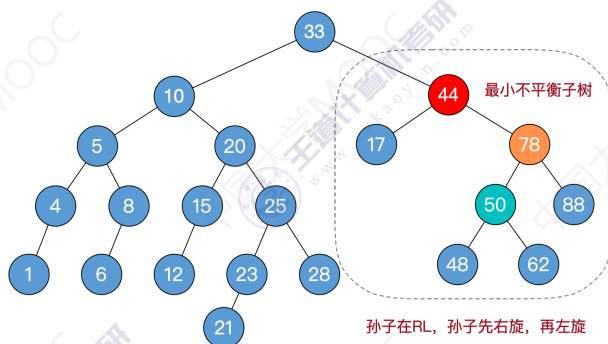
### ①删除结点（方法同“二叉排序树”）

### ②一路向北找到最小不平衡子树，找不到就完结撒花

③找最小不平衡子树下，“个头”最高的儿子、孙子

④根据孙子的位置，调整平衡（LL/RR/LR/RL）

⑤如果不平衡向上传导，继续②



王道考研/CSKAOYAN.COM

24

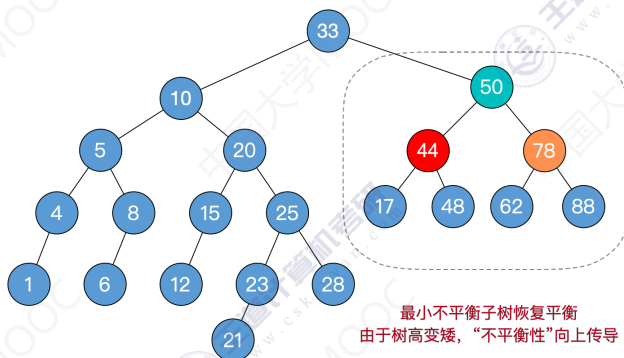
## AVL树删除操作——例4

平衡二叉树的**删除**操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
- ②一路向北找到最小不平衡子树，找不到就完结撒花
- ③找最小不平衡子树下，“个头”最高的儿子、孙子
- ④根据孙子的位置，调整平衡（LL/RR/LR/RL）

⑤如果不平衡向上传导，继续②

- 对最小不平衡子树的旋转可能导致树变矮，从而导致上层祖先不平衡（不平衡向上传递）



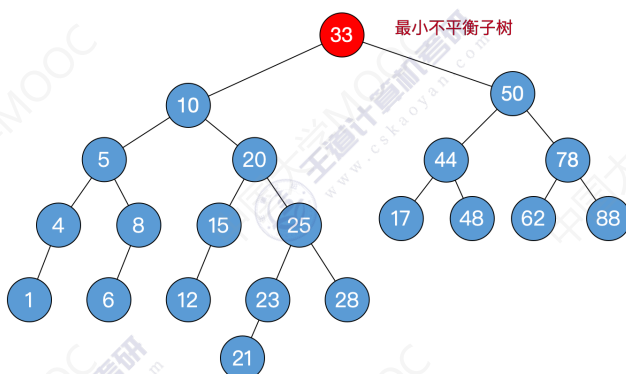
王道考研/CSKAOYAN.COM

25

## AVL树删除操作——例4

平衡二叉树的**删除**操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
- ②一路向北找到最小不平衡子树，找不到就完结撒花
- ③找最小不平衡子树下，“个头”最高的儿子、孙子
- ④根据孙子的位置，调整平衡（LL/RR/LR/RL）
- ⑤如果不平衡向上传导，继续②



王道考研/CSKAOYAN.COM

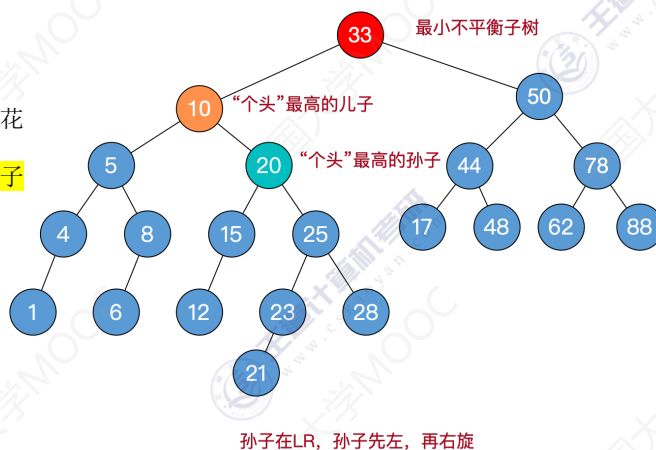
26



## AVL树删除操作——例4

平衡二叉树的删除操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
- ②一路向北找到最小不平衡子树，找不到就完结撒花
- ③找最小不平衡子树下，“个头”最高的儿子、孙子
- ④根据孙子的位置，调整平衡（LL/RR/LR/RL）
- ⑤如果不平衡向上传导，继续②



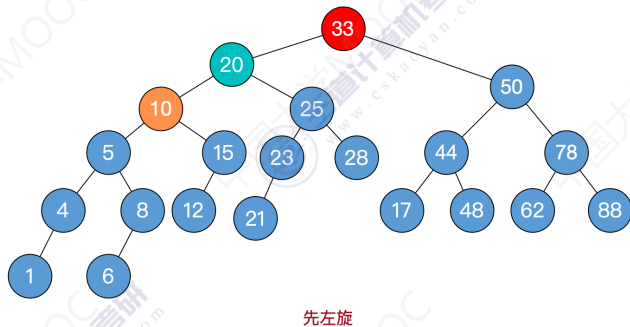
王道考研/CSKAOYAN.COM

27

## AVL树删除操作——例4

平衡二叉树的删除操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
- ②一路向北找到最小不平衡子树，找不到就完结撒花
- ③找最小不平衡子树下，“个头”最高的儿子、孙子
- ④根据孙子的位置，调整平衡（LL/RR/LR/RL）
  - 孙子在LL：儿子右单旋
  - 孙子在RR：儿子左单旋
  - 孙子在LR：孙子先左旋，再右旋
  - 孙子在RL：孙子先右旋，再左旋
- ⑤如果不平衡向上传导，继续②



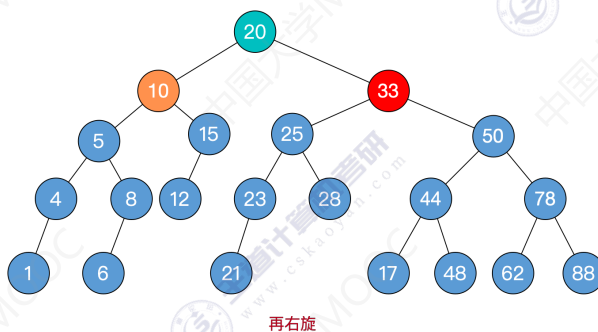
王道考研/CSKAOYAN.COM

28

## AVL树删除操作——例4

平衡二叉树的**删除**操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
- ②一路向北找到最小不平衡子树，找不到就完结撒花
- ③找最小不平衡子树下，“个头”最高的儿子、孙子
- ④根据孙子的位置，调整平衡（LL/RR/LR/RL）
  - 孙子在LL：儿子右单旋
  - 孙子在RR：儿子左单旋
  - 孙子在LR：孙子先左旋，再右旋
  - 孙子在RL：孙子先右旋，再左旋
- ⑤如果不平衡向上传导，继续②



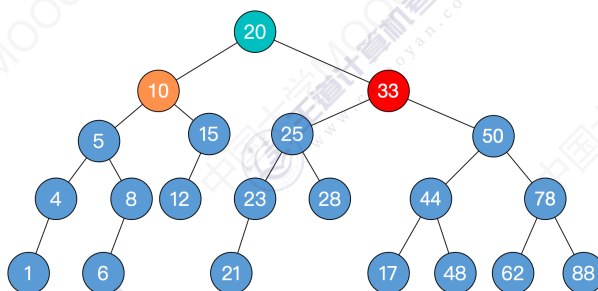
王道考研/CSKAOYAN.COM

29

## AVL树删除操作——例4

平衡二叉树的**删除**操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
- ②一路向北找到最小不平衡子树，找不到就完结撒花
- ③找最小不平衡子树下，“个头”最高的儿子、孙子
- ④根据孙子的位置，调整平衡（LL/RR/LR/RL）
- ⑤如果不平衡向上传导，继续②
  - 对最小不平衡子树的旋转可能导致树变矮，从而导
  - 致上层祖先不平衡（不平衡向上传递）



Over !



王道考研/CSKAOYAN.COM

30

### AVL树删除操作——例5

平衡二叉树的删除操作具体步骤：

①删除结点（方法同“二叉排序树”）

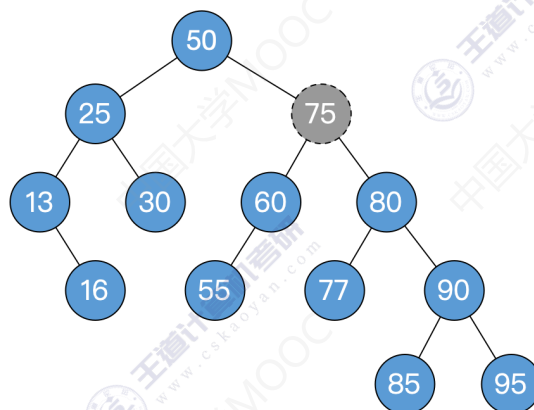
- 若删除的结点是叶子，直接删。
- 若删除的结点只有一个子树，用子树顶替删除位置
- 若删除的结点有两棵子树，用前驱（或后继）结点顶替，并转换为对前驱（或后继）结点的删除。

②一路向北找到最小不平衡子树，找不到就完结撒花

③找最小不平衡子树下，“个头”最高的儿子、孙子

④根据孙子的位置，调整平衡（LL/RR/LR/RL）

⑤如果不平衡向上传导，继续②



被删除结点有左右子树，用前驱结点顶替（复制数据即可）  
并转化为对前驱结点的删除

王道考研/CSKAOYAN.COM

31

### AVL树删除操作——例5

平衡二叉树的删除操作具体步骤：

①删除结点（方法同“二叉排序树”）

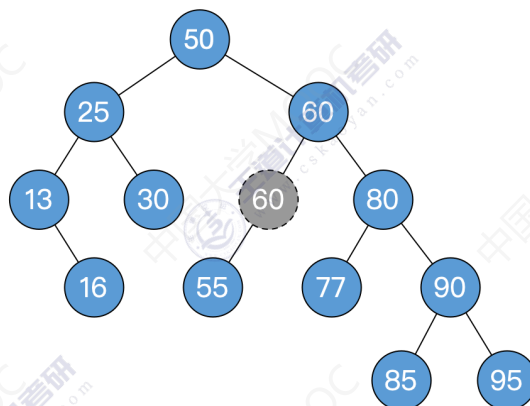
- 若删除的结点是叶子，直接删。
- 若删除的结点只有一个子树，用子树顶替删除位置
- 若删除的结点有两棵子树，用前驱（或后继）结点顶替，并转换为对前驱（或后继）结点的删除。

②一路向北找到最小不平衡子树，找不到就完结撒花

③找最小不平衡子树下，“个头”最高的儿子、孙子

④根据孙子的位置，调整平衡（LL/RR/LR/RL）

⑤如果不平衡向上传导，继续②



被删除结点只有左子树，用子树顶替删除位置（用结点实体顶替）

王道考研/CSKAOYAN.COM

32

### AVL树删除操作——例5

平衡二叉树的删除操作具体步骤：

①删除结点（方法同“二叉排序树”）

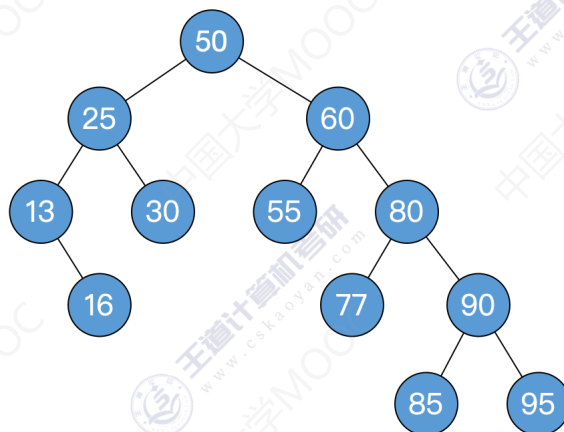
- 若删除的结点是叶子，直接删。
- 若删除的结点只有一个子树，用子树顶替删除位置
- 若删除的结点有两棵子树，用前驱（或后继）结点顶替，并转换为对前驱（或后继）结点的删除。

②一路向北找到最小不平衡子树，找不到就完结撒花

③找最小不平衡子树下，“个头”最高的儿子、孙子

④根据孙子的位置，调整平衡（LL/RR/LR/RL）

⑤如果不平衡向上传导，继续②



王道考研/CSKAOYAN.COM

33

### AVL树删除操作——例5

平衡二叉树的删除操作具体步骤：

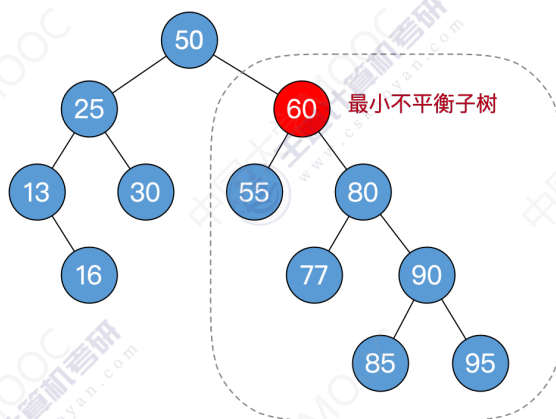
①删除结点（方法同“二叉排序树”）

②一路向北找到最小不平衡子树，找不到就完结撒花

③找最小不平衡子树下，“个头”最高的儿子、孙子

④根据孙子的位置，调整平衡（LL/RR/LR/RL）

⑤如果不平衡向上传导，继续②



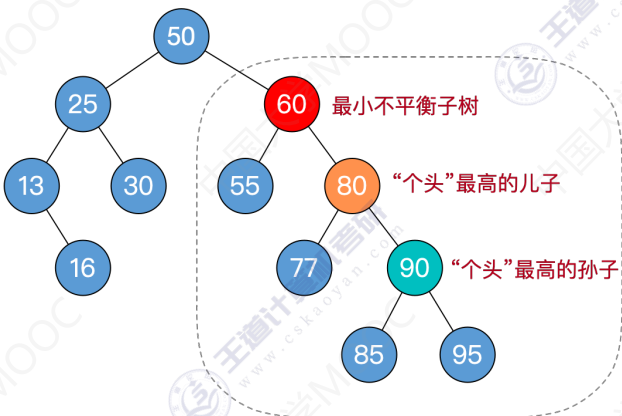
王道考研/CSKAOYAN.COM

34

## AVL树删除操作——例5

平衡二叉树的删除操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
- ②一路向北找到最小不平衡子树，找不到就完结撒花
- ③找最小不平衡子树下，“个头”最高的儿子、孙子
- ④根据孙子的位置，调整平衡（LL/RR/LR/RL）
- ⑤如果不平衡向上传导，继续②



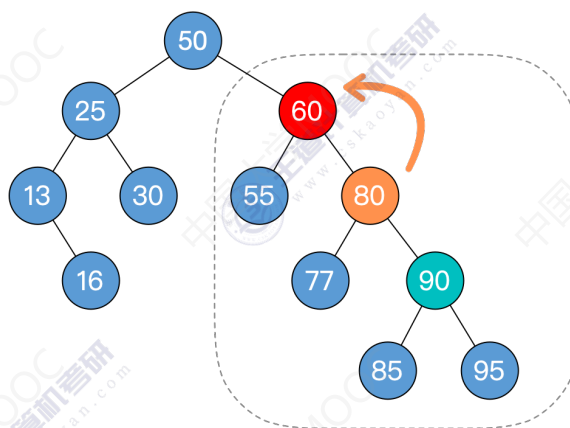
王道考研/CSKAOYAN.COM

35

## AVL树删除操作——例5

平衡二叉树的删除操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
- ②一路向北找到最小不平衡子树，找不到就完结撒花
- ③找最小不平衡子树下，“个头”最高的儿子、孙子
- ④根据孙子的位置，调整平衡（LL/RR/LR/RL）
  - 孙子在LL：儿子右单旋
  - 孙子在RR：儿子左单旋
  - 孙子在LR：孙子先左旋，再右旋
  - 孙子在RL：孙子先右旋，再左旋
- ⑤如果不平衡向上传导，继续②



孙子在RR，儿子左单旋

王道考研/CSKAOYAN.COM

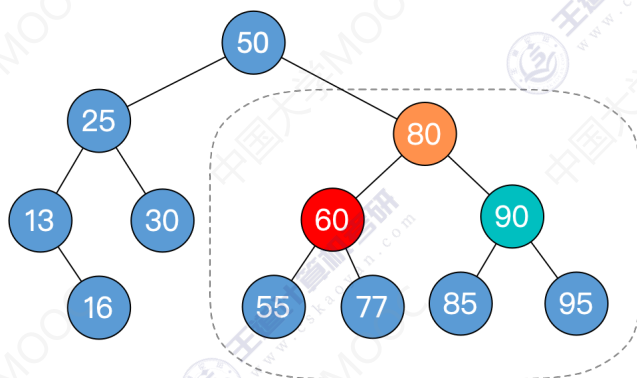
36



### AVL树删除操作——例5

平衡二叉树的删除操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
  - ②一路向北找到最小不平衡子树，找不到就完结撒花
  - ③找最小不平衡子树下，“个头”最高的儿子、孙子
  - ④根据孙子的位置，调整平衡（LL/RR/LR/RL）
  - ⑤如果不平衡向上传导，继续②
- 对最小不平衡子树的旋转可能导致树变矮，从而导致上层祖先不平衡（不平衡向上传递）



Over !



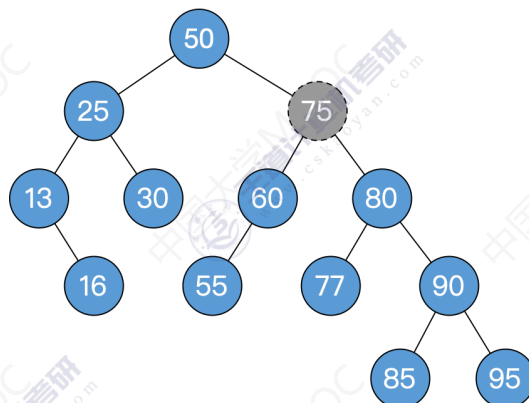
王道考研/CSKAOYAN.COM

37

### AVL树删除操作——例6

平衡二叉树的删除操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
- 若删除的结点是叶子，直接删。
  - 若删除的结点只有一个子树，用子树顶替删除位置
  - 若删除的结点有两棵子树，用前驱（或后继）结点顶替，并转换为对前驱（或后继）结点的删除。
- ②一路向北找到最小不平衡子树，找不到就完结撒花
  - ③找最小不平衡子树下，“个头”最高的儿子、孙子
  - ④根据孙子的位置，调整平衡（LL/RR/LR/RL）
  - ⑤如果不平衡向上传导，继续②



被删除结点有左右子树，用后继结点顶替（复制数据即可）  
并转化为对后继结点的删除

王道考研/CSKAOYAN.COM

38

### AVL树删除操作——例6

平衡二叉树的删除操作具体步骤：

①删除结点（方法同“二叉排序树”）

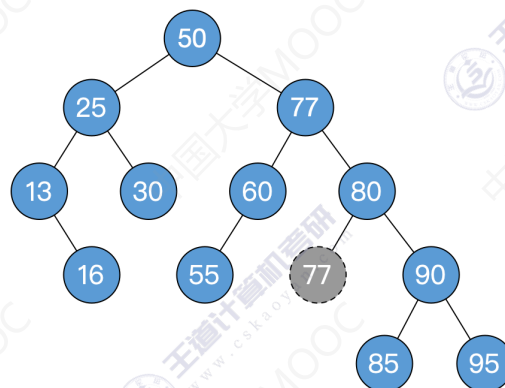
- 若删除的结点是叶子，直接删。
- 若删除的结点只有一个子树，用子树顶替删除位置
- 若删除的结点有两棵子树，用前驱（或后继）结点顶替，并转换为对前驱（或后继）结点的删除。

②一路向北找到最小不平衡子树，找不到就完结撒花

③找最小不平衡子树下，“个头”最高的儿子、孙子

④根据孙子的位置，调整平衡（LL/RR/LR/RL）

⑤如果不平衡向上传导，继续②



被删除结点为叶子，直接删即可

王道考研/CSKAOYAN.COM

39

### AVL树删除操作——例6

平衡二叉树的删除操作具体步骤：

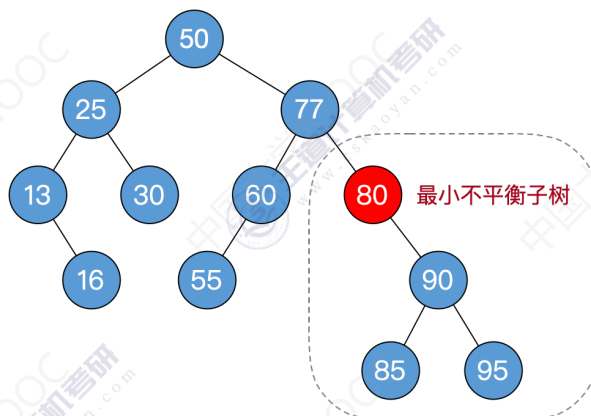
①删除结点（方法同“二叉排序树”）

②一路向北找到最小不平衡子树，找不到就完结撒花

③找最小不平衡子树下，“个头”最高的儿子、孙子

④根据孙子的位置，调整平衡（LL/RR/LR/RL）

⑤如果不平衡向上传导，继续②



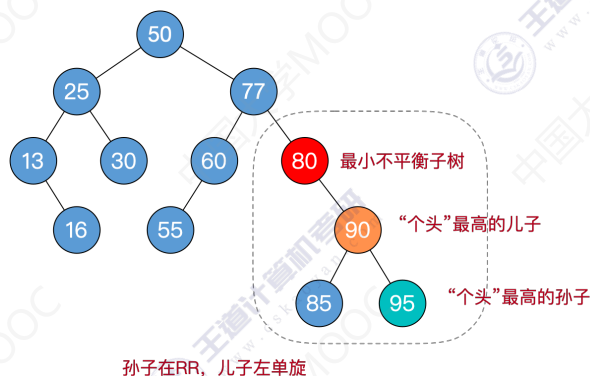
王道考研/CSKAOYAN.COM

40

## AVL树删除操作——例6

平衡二叉树的删除操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
- ②一路向北找到最小不平衡子树，找不到就完结撒花
- ③找最小不平衡子树下，“个头”最高的儿子、孙子
- ④根据孙子的位置，调整平衡（LL/RR/LR/RL）
- ⑤如果不平衡向上传导，继续②



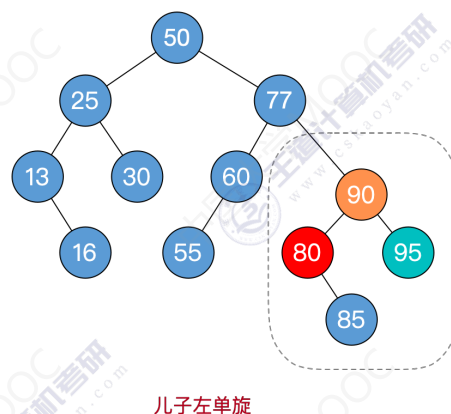
王道考研/CSKAOYAN.COM

41

## AVL树删除操作——例6

平衡二叉树的删除操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
- ②一路向北找到最小不平衡子树，找不到就完结撒花
- ③找最小不平衡子树下，“个头”最高的儿子、孙子
- ④根据孙子的位置，调整平衡（LL/RR/LR/RL）
  - 孙子在LL：儿子右单旋
  - 孙子在RR：儿子左单旋
  - 孙子在LR：孙子先左旋，再右旋
  - 孙子在RL：孙子先右旋，再左旋
- ⑤如果不平衡向上传导，继续②



王道考研/CSKAOYAN.COM

42

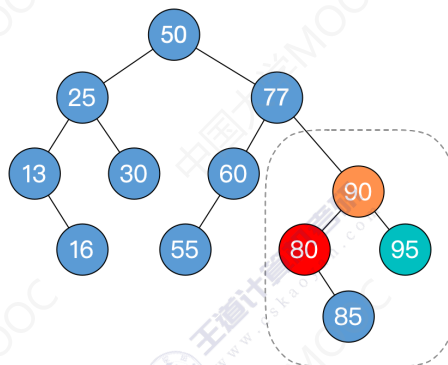
## AVL树删除操作——例6

平衡二叉树的删除操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
- ②一路向北找到最小不平衡子树，找不到就完结撒花
- ③找最小不平衡子树下，“个头”最高的儿子、孙子
- ④根据孙子的位置，调整平衡（LL/RR/LR/RL）

⑤如果不平衡向上传导，继续②

- 对最小不平衡子树的旋转可能导致树变矮，从而导致上层祖先不平衡（不平衡向上传递）



Over !



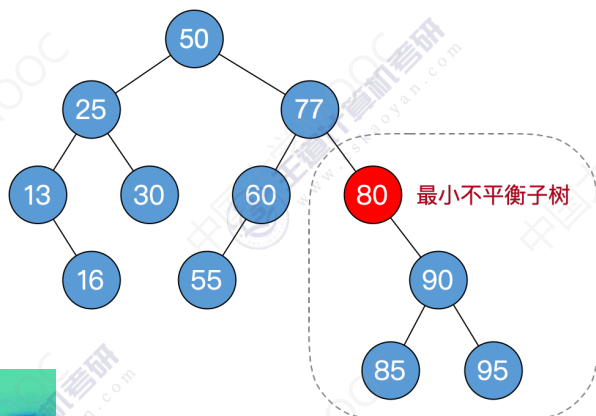
王道考研/CSKAOYAN.COM

43

## AVL树删除操作——例6

平衡二叉树的删除操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
- ②一路向北找到最小不平衡子树，找不到就完结撒花
- ③找最小不平衡子树下，“个头”最高的儿子、孙子
- ④根据孙子的位置，调整平衡（LL/RR/LR/RL）
- ⑤如果不平衡向上传导，继续②



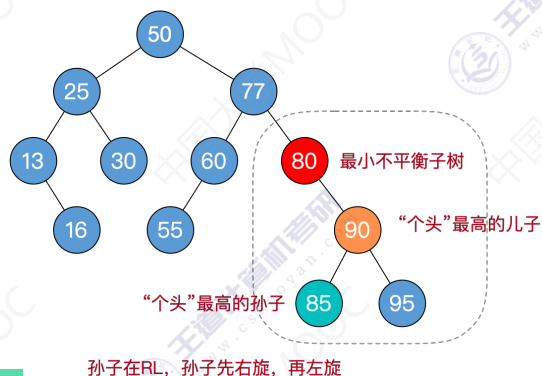
王道考研/CSKAOYAN.COM

44

## AVL树删除操作——例6

平衡二叉树的删除操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
- ②一路向北找到最小不平衡子树，找不到就完结撒花
- ③找最小不平衡子树下，“个头”最高的儿子、孙子
- ④根据孙子的位置，调整平衡（LL/RR/LR/RL）
- ⑤如果不平衡向上传导，继续②



有个问题

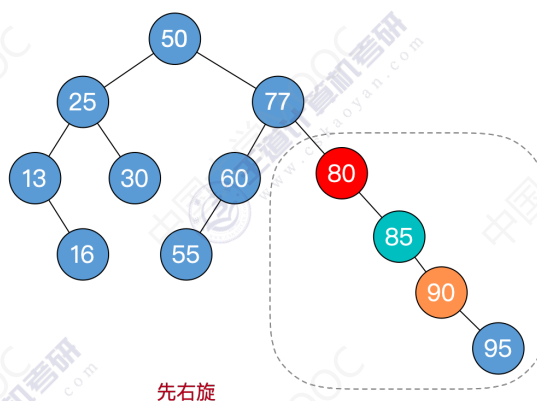
王道考研/CSKAOYAN.COM

45

## AVL树删除操作——例6

平衡二叉树的删除操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
- ②一路向北找到最小不平衡子树，找不到就完结撒花
- ③找最小不平衡子树下，“个头”最高的儿子、孙子
- ④根据孙子的位置，调整平衡（LL/RR/LR/RL）
  - 孙子在LL：儿子右单旋
  - 孙子在RR：儿子左单旋
  - 孙子在LR：孙子先左旋，再右旋
  - 孙子在RL：孙子先右旋，再左旋
- ⑤如果不平衡向上传导，继续②



王道考研/CSKAOYAN.COM

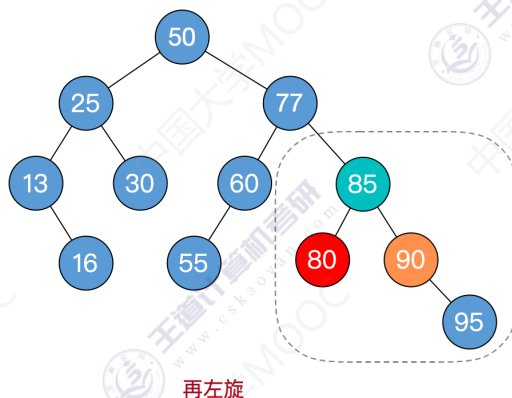
46



## AVL树删除操作——例6

平衡二叉树的删除操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
- ②一路向北找到最小不平衡子树，找不到就完结撒花
- ③找最小不平衡子树下，“个头”最高的儿子、孙子
- ④根据孙子的位置，调整平衡（LL/RR/LR/RL）
  - 孙子在LL：儿子右单旋
  - 孙子在RR：儿子左单旋
  - 孙子在LR：孙子先左旋，再右旋
  - 孙子在RL：孙子先右旋，再左旋
- ⑤如果不平衡向上传导，继续②



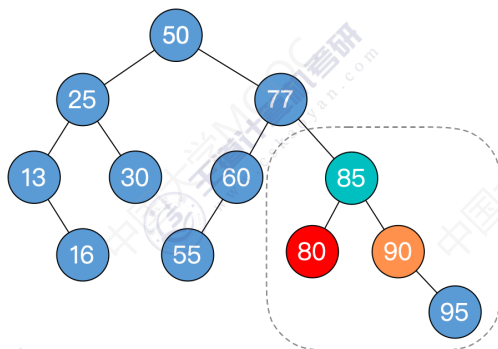
王道考研/CSKAOYAN.COM

47

## AVL树删除操作——例6

平衡二叉树的删除操作具体步骤：

- ①删除结点（方法同“二叉排序树”）
- ②一路向北找到最小不平衡子树，找不到就完结撒花
- ③找最小不平衡子树下，“个头”最高的儿子、孙子
- ④根据孙子的位置，调整平衡（LL/RR/LR/RL）
- ⑤如果不平衡向上传导，继续②
  - 对最小不平衡子树的旋转可能导致树变矮，从而导致上层祖先不平衡（不平衡向上传递）



Over !



不可能考这种有多种处理方式的题目！

王道考研/CSKAOYAN.COM

48

## 知识回顾与重要考点

平衡二叉树的删除操作具体步骤:

①删除结点（方法同“二叉排序树”）

- 若删除的结点是叶子，直接删。
- 若删除的结点只有一个子树，用子树顶替删除位置
- 若删除的结点有两棵子树，用前驱（或后继）结点顶替，并转换为对前驱（或后继）结点的删除。

②一路向北找到最小不平衡子树，找不到就完结撒花

③找最小不平衡子树下，“个头”最高的儿子、孙子

④根据孙子的位置，调整平衡（LL/RR/LR/RL）

- 孙子在LL：儿子右单旋
- 孙子在RR：儿子左单旋
- 孙子在LR：孙子先左旋，再右旋
- 孙子在RL：孙子先右旋，再左旋

平衡二叉树删除操作时间复杂度= $O(\log_2 n)$

⑤如果不平衡向上传导，继续②

- 对最小不平衡子树的旋转可能导致树变矮，从而导致上层祖先不平衡（不平衡向上传递）

王道考研/CSKAOYAN.COM