

数据结构

查找（二）

深圳大学电子与信息工程学院 周飞

2 第七章 查找

第三节 动态查找表

- 一、动态查找表
 - 表本身是在查找过程中动态生成的。
 - 若表中存在着其关键字等于查找关键字key的记录，则查找成功；
 - 否则在表中插入关键字为key的记录。

3 第七章 查找

第三节 动态查找表

- 二、二叉排序树（定义）

- 二叉排序树是：

- （1）空树或满足以下条件的二叉树：

- （2）考察树中任意一个结点，

- (2-1)若其左子树不为空，则左子树上的所有结点的关键字都小于根的关键字；

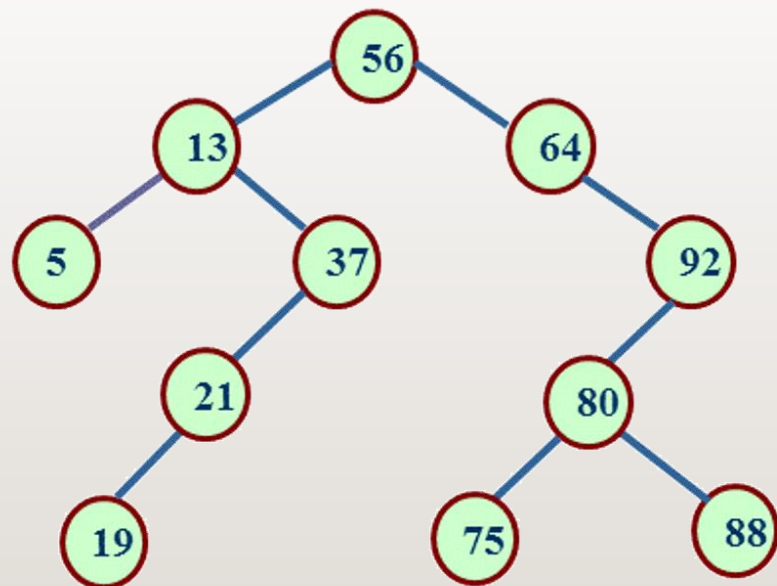
- (2-2)若其右子树不为空，则右子树上的所有结点的关键字都大于根的关键字；

- (2-3)其左、右子树也都是二叉排序树。

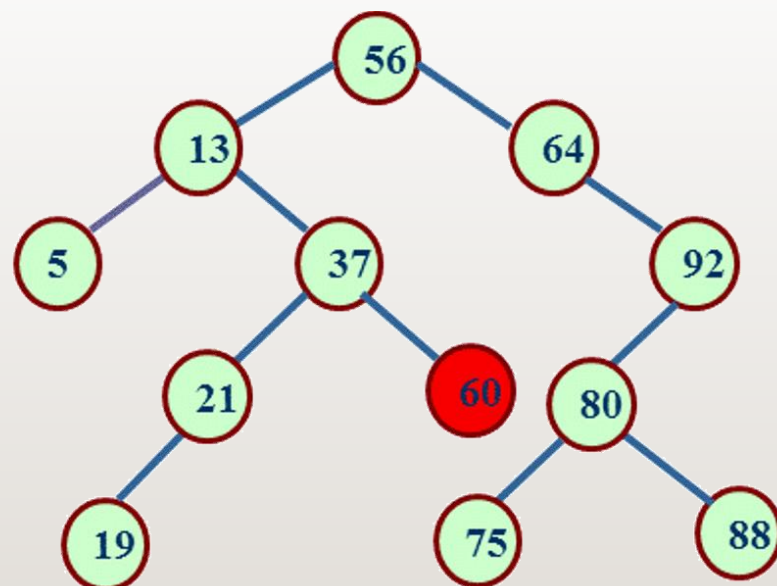
4 第七章 查找

第三节 动态查找表

- 二、二叉排序树



二叉排序树



非二叉排序树

5 第七章 查找

第三节 动态查找表

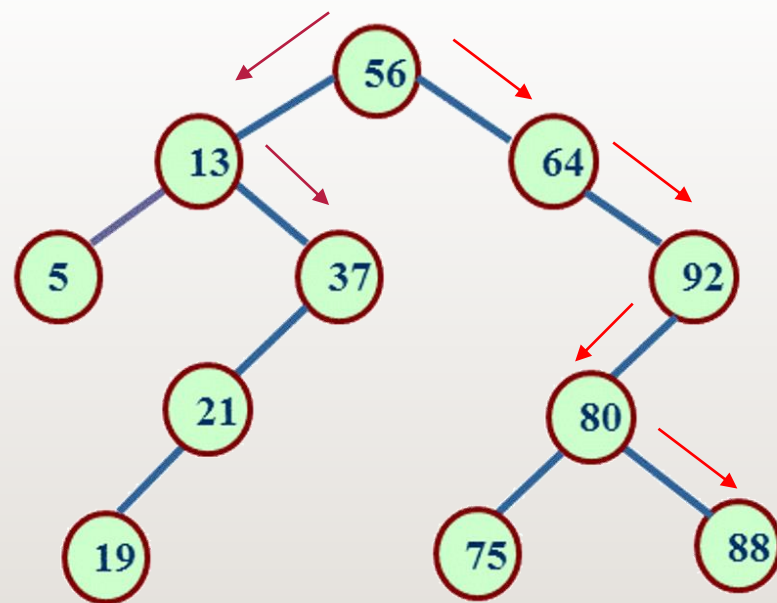
- 二、二叉排序树（查找）

二叉排序树又称二叉查找树

- 查找算法：

给定值与根结点比较：

1. 若相等，查找成功
2. 若小于，查找左子树
3. 若大于，查找右子树



在二叉排序树中查找关键字值等于37, 88

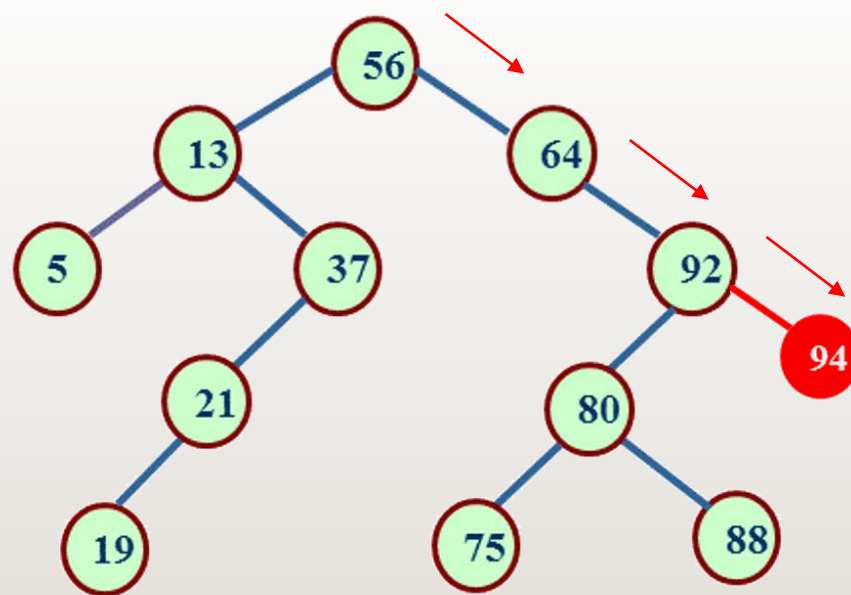
6 第七章 查找

第三节 动态查找表

- 二、二叉排序树（插入）

查找94

插入94



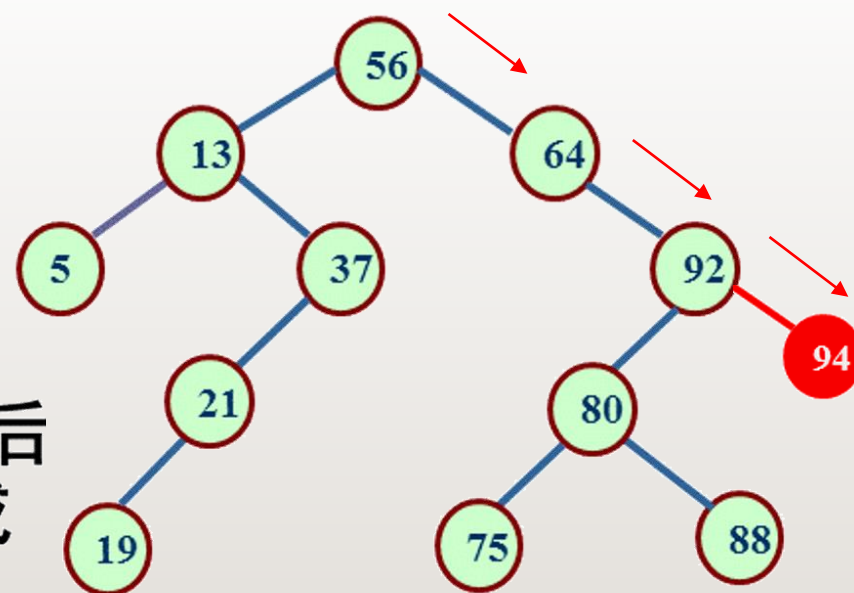
7 第七章 查找

第三节 动态查找表

- 二、二叉排序树（插入）

新插入的结点一定是叶子结点（只需改动一个结点的指针）

该叶子结点是查找不成功时路径上访问的最后一个结点的左孩子或右孩子（新结点值小于或大于该结点值）

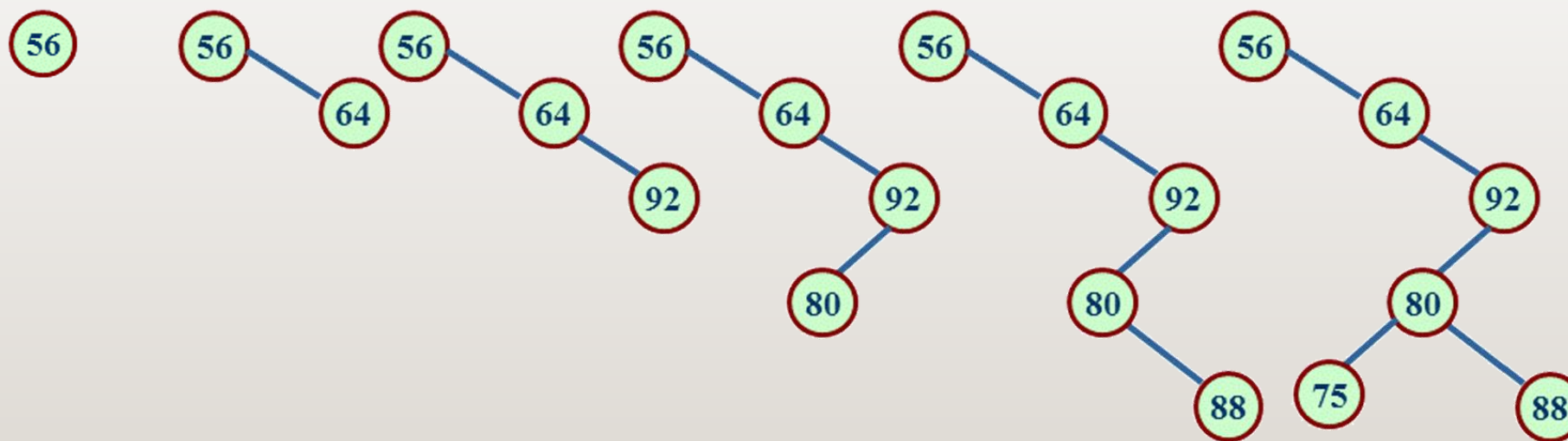


8 第七章 查找

第三节 动态查找表

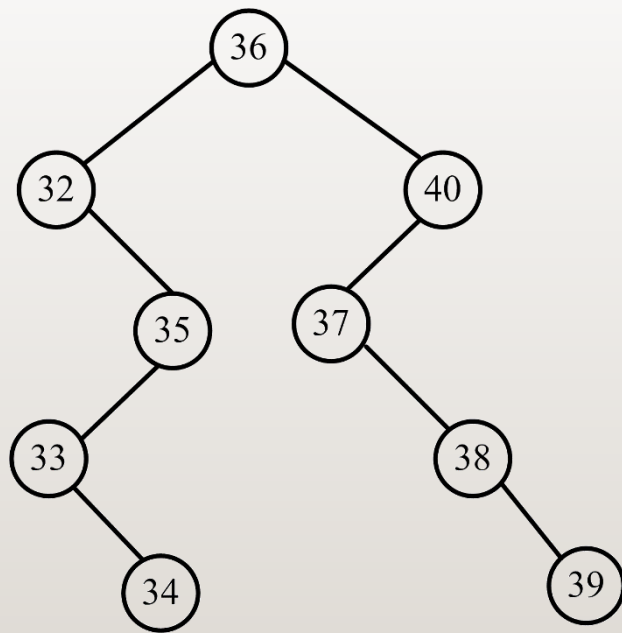
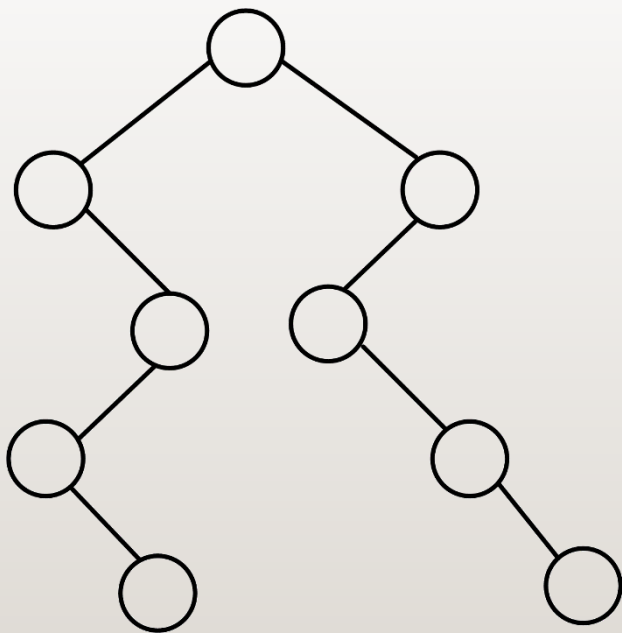
- 二、二叉排序树（生成）

画出在初始为空的二叉排序树中依次插入
56, 64, 92, 80, 88, 75时该树的生长全过程



9

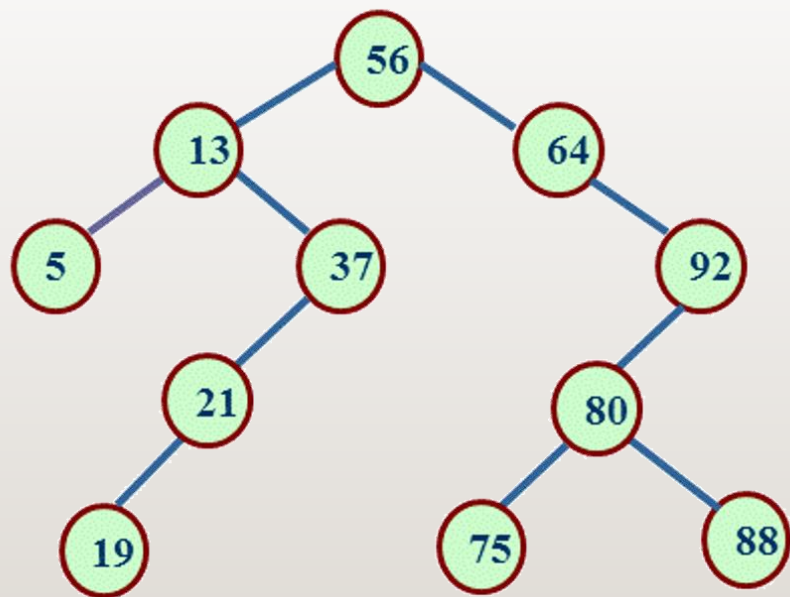
3. 二叉排序树的结构如下图所示，其中各结点的关键字依次为32-40，请标出各节点的关键字并写出求解过程。



10 第七章 查找

第三节 动态查找表

- 二、二叉排序树



得到一个关键字的有序序列

5, 13, 19, 21, 37, 56, 64, 75, 80, 88, 92

中序遍历二叉排序树

11 第七章 查找

第三节 动态查找表

- 二、二叉排序树（删除）
 - 要求：
 - 删除二叉排序树中的一个结点后，必须保持二叉排序树的特性（左子树的所有结点值小于根结点，右子树的所有结点值大于根结点）
 - 也即保持中序遍历后，输出为有序序列

12 第七章 查找

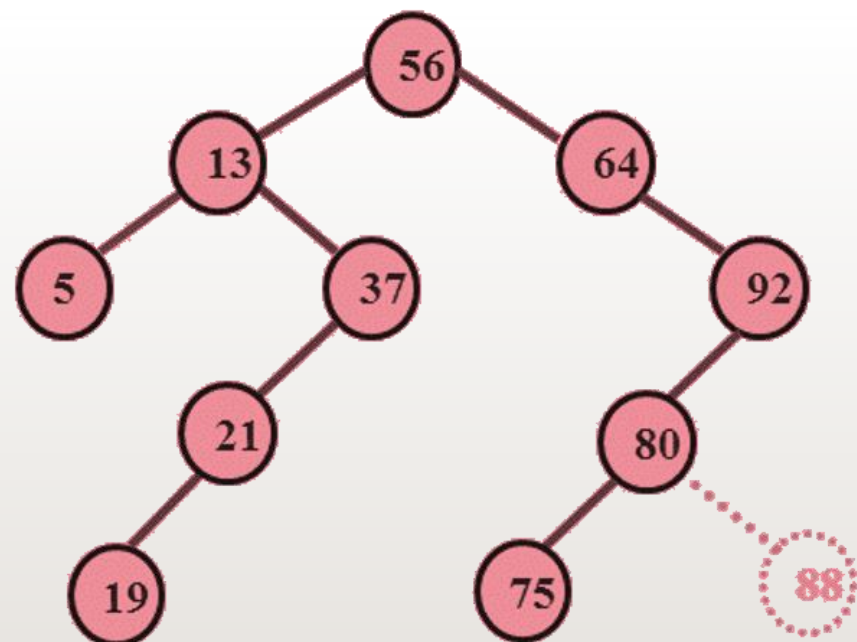
第三节 动态查找表

- 二、二叉排序树（删除）
 - 被删除结点具有以下三种情况：
 1. 是叶子结点
 2. 只有左子树或右子树
 3. 同时有左、右子树

13 第七章 查找

第三节 动态查找表

- 二、二叉排序树（删除）
 - 1. 被删除结点叶子结点
 - 直接删除结点，让其双亲结点指向它的指针变为NULL

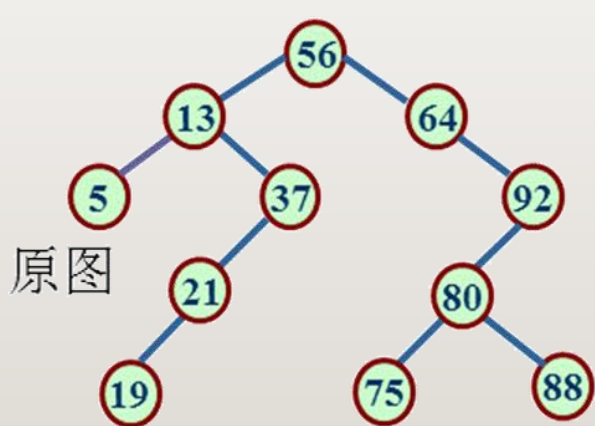


删除结点88

14 第七章 查找

第三节 动态查找表

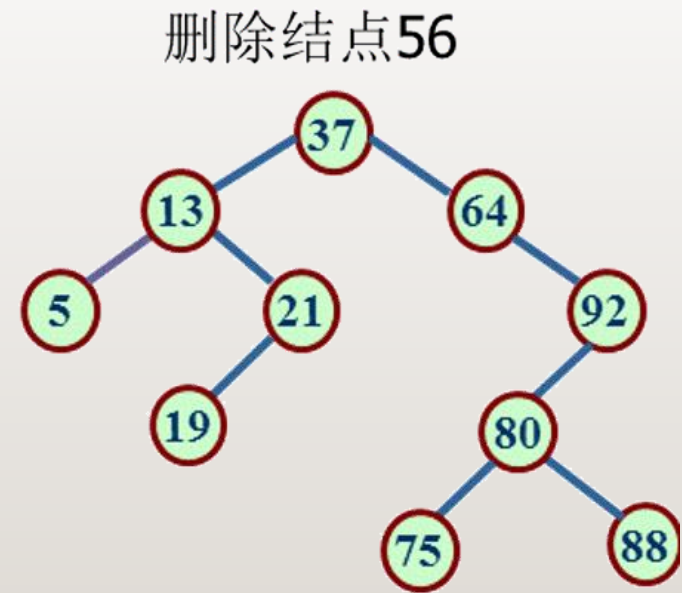
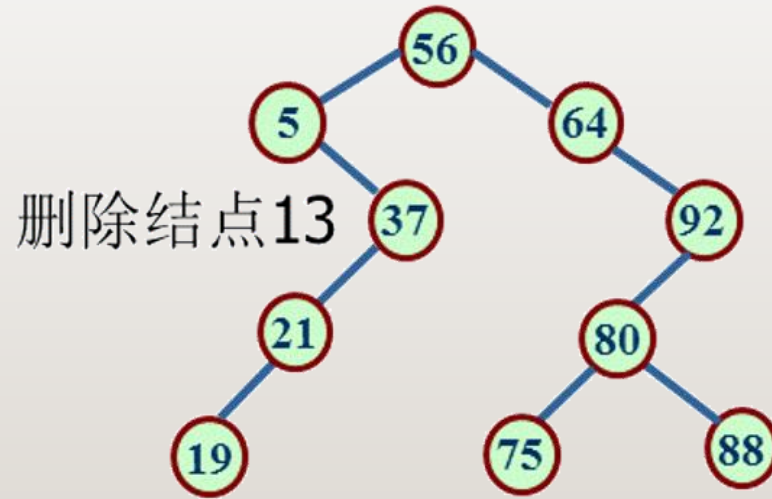
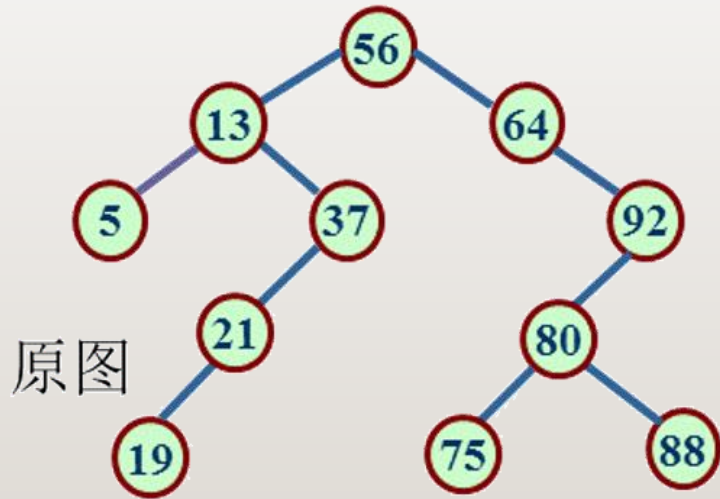
- 二、二叉排序树（删除）
 - 2. 被删除结点只有左子树或者右子树
 - 删除结点，让其双亲结点指向其左孩子或右孩子，即用它的孩子替代它。



15 第七章 查找

第三节 动态查找表

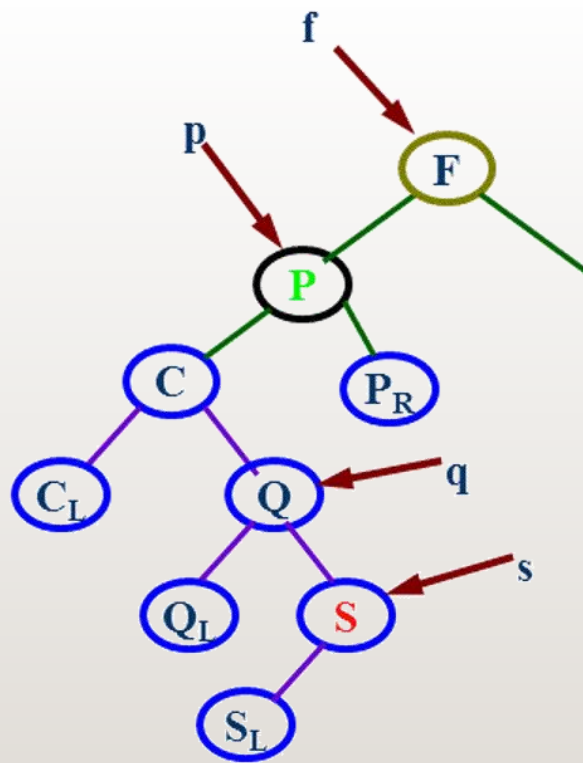
- 二、二叉排序树（删除）
 - 3. 被删除结点p既有左子树又有右子树（举例）



16 第七章 查找

第三节 动态查找表

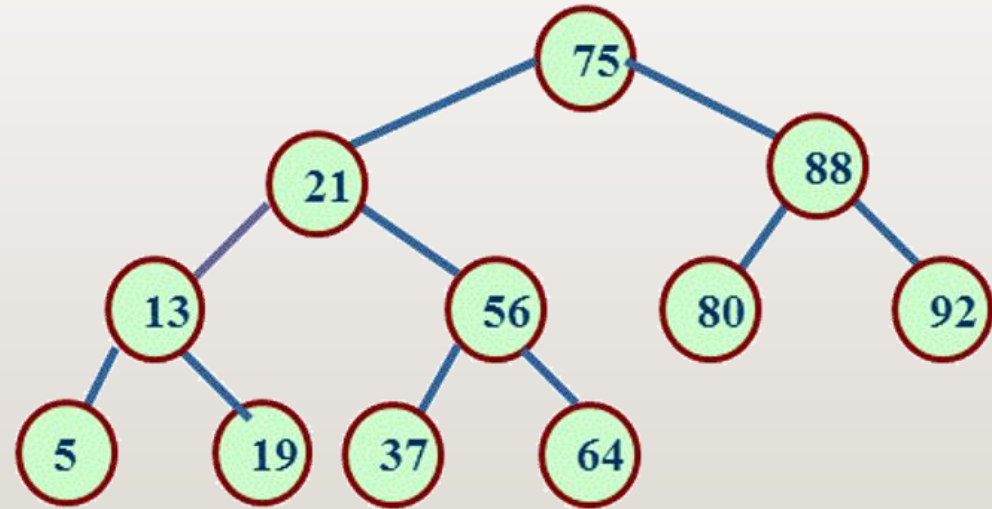
- 二、二叉排序树（删除）
 - 3. 被删除结点 p 既有左子树又有右子树
 - 以中序遍历时的直接前驱 s 替代被删除结点 p ，然后再删除该直接前驱（只可能有左孩子）



17 第七章 查找

第三节 动态查找表

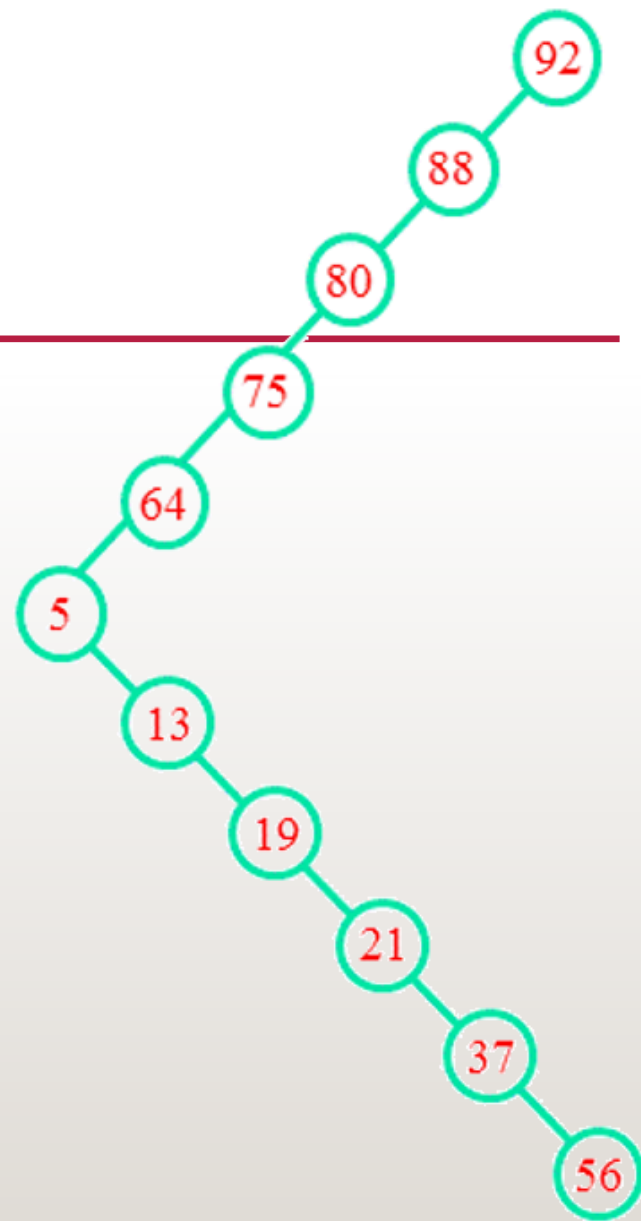
- 二、二叉排序树（性能分析）
 - 在最好的情况下，二叉排序树可以近似于一个完全二叉树，其时间复杂度为 $O(\log_2 n)$



18 第七章 查找

第三节 动态查找表

- 二、二叉排序树（性能分析）
 - 在最坏的情况下，
二叉排序树可以近似于一个线性表，
其时间复杂度为 $O(n)$



19 第七章 查找

第三节 动态查找表

- 二、二叉排序树（特性）
 - 一个无序序列可以通过构造一棵二叉排序树而变成一个有序序列（通过中序遍历）
 - 二叉排序树既拥有类似于折半查找的特性，又采用了链表作存储结构
 - 插入新记录时，只需改变一个结点的指针，相当于在有序序列中插入一个记录而不需要移动其它记录

20 第七章 查找

第三节 动态查找表

- 二、二叉排序树（特性）
 - 但当插入记录的次序不当时(如升序或降序)，则二叉排序树深度很深(n)，增加了查找的时间

