

# 二叉排序树 (BST)

---

## 一、基本介绍

二叉排序树：BST：(Binary Sort(Search) Tree)，对于二叉排序树的任何一个非叶子节点，要求左子节点的值比当前节点的值小，右子节点的值比当前节点的值大。

特别说明：如果有相同的值，可以将该节点放在左子节点或右子节点

## 二、二叉排序树的删除

二叉排序树的删除情况比较复杂，有下面三种情况需要考虑：

### 第一种情况:删除叶子节点

- (1) 需求先去找到要删除的结点 `targetNode`
- (2) 找到 `targetNode` 的 父结点 `parent`
- (3) 确定 `targetNode` 是 `parent`的左子结点 还是右子结点
- (4) 根据前面的情况来对应删除

左子结点 `parent.left = null`

右子结点 `parent.right = null;`

### 第二种情况: 删除只有一颗子树的节点

- (1) 需求先去找到要删除的结点 `targetNode`
- (2) 找到 `targetNode` 的 父结点 `parent`
- (3) 确定 `targetNode` 的子结点是左子结点还是右子结点
- (4) `targetNode` 是 `parent` 的左子结点还是右子结点
- (5) .判断Parent是否为空，空的话：

`root = targetNode.left;`

- 1、如果 `targetNode` 是 `parent` 的左子结点

`parent.left = targetNode.left;`

2、如果 targetNode 是 parent 的右子结点

parent.right = targetNode.left;

(6) 如果 targetNode 有右子结点

1、如果 targetNode 是 parent 的左子结点

parent.left = targetNode.right;

2、如果 targetNode 是 parent 的右子结点

parent.right = targetNode.right

### **情况三：删除有两颗子树的节点.**

(1) 需求先去找到要删除的结点 targetNode

(2) 找到 targetNode 的父结点 parent

(3) 从 targetNode 的右子树找到最小的结点

(4) 用一个临时变量，将最小结点的值保存 temp = 11

(5) 删除该最小结点

(6) targetNode.value = temp