

# 二叉搜索树删除操作

(delete\_bst.cpp/c/pas)

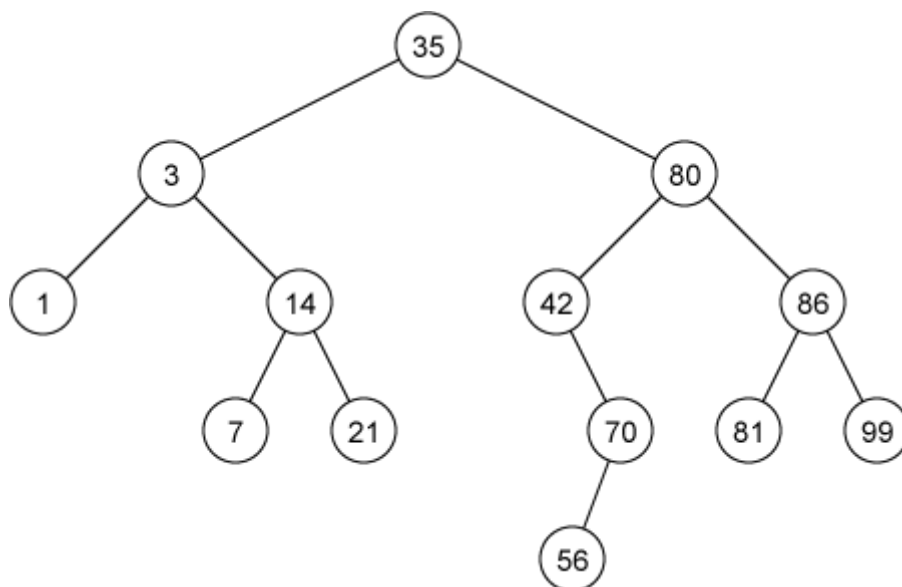
Time Limit : 2 sec , Memory Limit : 128MB

## Description

搜索树是一种可以进行插入、搜索、删除等操作的数据结构，可以用作字典或优先级队列。二叉搜索树属于最基本的搜索树。二叉搜索树的各结点均拥有键值，且满足下述性质：

- 设  $x$  为二叉搜索树的结点。如果  $y$  是  $x$  左子树中的结点，那么  $y$  的键值  $y.key \leq x.key$ 。如果  $y$  是  $x$  右子树中的结点，那么  $x$  的键值  $x.key \leq y.key$ 。

下图是二叉搜索树的例子。



你的任务是编写一个程序，能在二叉搜索树  $T$  上完成以下命令：

- **insert  $k$ :** 插入一个键值为  $k$  的结点。
- **find  $k$ :** 在二叉搜索树  $T$  上查找含有键值  $k$  的结点。
- **delete  $k$ :** 删除一个键值为  $k$  的结点。
- **print:** 输出中序遍历算法和前序遍历算法所得到的键值序列。

删除操作 **delete  $k$**  的具体方法参考如下：

1. 如果  $z$  没有儿子结点，修改  $z$  的父亲  $z \rightarrow p$  为 **NIL** (delete  $z$ )。
2. 如果  $z$  仅有一个儿子结点，直接连接  $z$  的父亲结点和  $z$  的孩子结点。
3. 如果  $z$  有两个孩子，则用  $z$  的中序遍历的直接前驱  $y$  来连接，并用  $y$  的键值代替  $z$  的键值。

## Input (delete\_bst.in)

第一行输入命令数  $m$ 。

接下来  $m$  行以 insert k, find k, delete k 或者 print 的格式输入命令，每个命令占 1 行。

## Output (delete\_bst.out)

每执行 1 次 find 命令后，若找到则输出 "yes"，否则输出 "no"。

每执行 1 次 print 命令后，就分别输出中序遍历算法和前序遍历算法所得到的键值序列，每个序列占 1 行。每个键值前输出 1 个空格。

## Constraints

- 命令数  $\leq 500,000$
- print 命令数  $\leq 10$
- $-2,000,000,000 \leq k \leq 2,000,000,000$
- 树的高度不会超过 100
- 二叉搜索树各结点的键值不重复

## Sample Input 1

```
18

insert 8

insert 2

insert 3

insert 7

insert 22

insert 1

find 1

find 2

find 3
```

```
find 4

find 5

find 6

find 7

find 8

print

delete 3

delete 7

print
```

## Sample Output 1

---

```
yes

yes

yes

no

no

no

yes

yes

1 2 3 7 8 22

8 2 1 3 7 22

1 2 8 22

8 2 1 22
```