

2022 年秋季-数据结构-实验 3

读取 N 个整数，基于这 N 个整数构造一棵平衡二叉搜索树。

平衡二叉树的定义为：二叉树的所有结点的左右子树的高度差的绝对值不超过 1。基本要求：

(1) 对读取的 N 个整数的数组，采用堆排序进行就地排序（即除了该数组外，仅使用 $O(1)$ 的额外空间）；

(2) 对排序好的数组，设计方法（可以采用递归）构造平衡二叉搜索树；

(3) 设计方法（可以采用递归）判断给定的一棵平衡二叉搜索树是否满足平衡的要求；

(4) 测试自己构造的二叉树是否是平衡的。

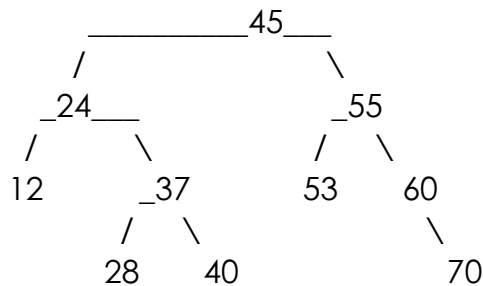
测试用例：

（第一行为 N ，第二行为空格分隔的 N 个 `int` 类型整数）

10

45 24 55 53 60 70 12 37 28 40

构造的平衡二叉树可为（可有其他的形式，只要能保证构造二叉树为平衡即可）：



提示：

考虑如下的解决第 2 个问题的思路。

N 个已排序好的整数，下标从 0 到 $N-1$ 。考虑下标为 $(N-1)/2$ 向下取整，定义其为 `center`。将 `center` 构造为 `root` 结点，左子树用递归去解决 0 到 `center-1`，右子树用递归去解决从 `center+1` 到 $N-1$ 。

正确性的理解如下：

考虑 N 的二进制表示形式的最高位，设其为 k （最低位为 0，例如 $11 = (1011)_2$ 的最高位为 3），则上述构造方法构造出来的二叉树的层数为 $k+1$ 。

采取归纳法去理解。 $N=0$ 时，最高位上的值为 0，最高位因此可理解为 -1，相应构造出来的二叉树的层数为 0，为空树。

$N>0$ 时，有两种情况：第 1 种， N 为奇数，则最低位上的值为 1，那么 0 到 `center-1` 有 $(N-1)/2$ 个整数，`center+1` 到 $N-1$ 也是如此。而 $(N-1)/2$ 的最高位为 $k-1$ （除以 2 相当于二进制形式的右移），所以对于根结点来说，其左右子树的高度差的绝对值为 0，且左右子树也是平衡的，可以通过归纳法去理解；

第 2 种， N 为偶数，则最低位上的值为 0，那么 0 到 `center-1` 有 $(N-2)/2$ 个整数；`center+1` 到 $N-1$ 有 $N/2$ 个整数。归纳去理解，可知左子树是平衡的，且其的层数为 $k-1$ ，或者 k ；而右子树的层数为 k 。因此左右子树的高度差的绝对值不超过 1。