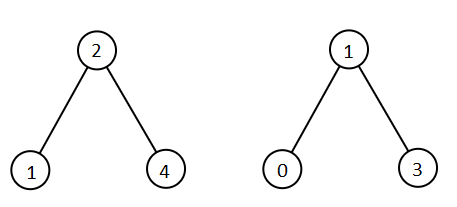
# 描述

给你 root1和root2这两棵二叉搜索树。

请你返回一个列表，其中包含两棵树中的所有整数并按升序排序。.

**示例 1：**



输入：root1 = [2,1,4], root2 = [1,0,3]

输出：[0,1,1,2,3,4]

**示例 2：**

输入：root1 = [0,-10,10], root2 = [5,1,7,0,2]

输出：[-10,0,0,1,2,5,7,10]

**示例 3：**

输入：root1 = [], root2 = [5,1,7,0,2]

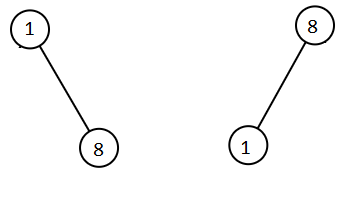
输出：[0,1,2,5,7]

**示例 4：**

输入：root1 = [0,-10,10], root2 = []

输出：[-10,0,10]

**示例 5：**



输入：root1 = [1,null,8], root2 = [8,1]

输出：[1,1,8,8]

**提示：**

每棵树最多有 5000 个节点。

每个节点的值在 [-10^5, 10^5] 之间。

# 分析

方法一：遍历 + 排序

我们可以想到的最简单的方法是，对两棵树进行任意形式的遍历（深度优先搜索、广度优先搜索、前序遍历、中序遍历、后序遍历），并将遍历到的所有元素放入一个数组中，最后对这个数组进行排序即可。

方法二：中序遍历 + 归并排序

方法一中并没有用到二叉搜索树本身的性质。如果我们对二叉搜索树进行中序遍历，就可以直接得到树中所有元素升序排序后的结果。因此我们可以对两棵树分别进行中序遍历，得到数组 v1 和 v2，它们分别存放了两棵树中的所有元素，且均已有序。在这之后，我们通过归并排序的方法对 v1 和 v2 进行排序，就可以得到最终的结果。

# 代码

## C

## C++

## Java

## Python