



链滴

# 二叉搜索树的后序遍历序列

作者: [yudake](#)

原文链接: <https://ld246.com/article/1518838362763>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

## 题目描述

输入一个整数数组，判断该数组是不是某二叉搜索树的后序遍历的结果。如果是则输出Yes,否则输出N。假设输入的数组的任意两个数字都互不相同。

## 解题思路

**已知条件：后序序列最后一个值为root；二叉搜索树左子树值都比root小，右子树值都比root大。**

- 确定root；
- 遍历序列（除去root结点），找到第一个大于root的位置，则该位置左边为左子树，右边为右子树；
- 遍历右子树，若发现有小于root的值，则直接返回false；
- 分别判断左子树和右子树是否仍是二叉搜索树（即递归步骤1、2、3）。

## 代码

```
public class Solution {
    public boolean VerifySequenceOfBST(int [] sequence) {
        if (sequence == null || sequence.length == 0)
            return false;
        return isPostBST(sequence, 0, sequence.length-1);
    }

    private boolean isPostBST(int[] sequence, int start, int end) {
        if (end <= start)
            return true;
        int i = start;
        for (; i < end; i++) {
            if (sequence[i] > sequence[end])
                break;
        }
        for (int j = i; j < end; j++) {
            if (sequence[j] < sequence[end])
                return false;
        }
        return isPostBST(sequence, start, i-1) && isPostBST(sequence, i, end-1);
    }
}
```