# 二叉搜索树的后序遍历序列

## 题目

[牛客网](https://www.nowcoder.com/practice/a861533d45854474ac791d90e447bafd?tpId=13&tqId=11176&rp=1&ru=%2Fta%2Fcoding-interviews&qru=%2Fta%2Fcoding-interviews%2Fquestion-ranking&tPage=1)

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 No 。假设输入的数组的任意两个数字都互不相同。

## 解题思路

1. 后序遍历中，最后一个节点为 root 节点
2. 由于 BST 的左子树都小于 root，右子树都大于 root，那么可以判定该节点是否为 BST
3. 依次类推，通过递归方式，再判定左右子树

public boolean VerifySquenceOfBST(int[] sequence) {  
 if (sequence.length == 0) {  
 return false;  
 }  
  
 if (sequence.length == 1) {  
 return true;  
 }  
  
 return isBST(sequence, 0, sequence.length - 1);  
}  
  
private boolean isBST(int[] sequence, int start, int end) {  
 if (start < 0 || end < 0 || start >= end) {  
 return true;  
 }  
  
 int rootV = sequence[end];  
 int rightIndex = -1, rightV = Integer.MIN\_VALUE;  
 for (int i = start; i < end; i++) {  
 if (rightV == Integer.MIN\_VALUE && sequence[i] > rootV) {  
 rightV = sequence[i];  
 rightIndex = i;  
 continue;  
 }  
  
 if (rightV != Integer.MIN\_VALUE && sequence[i] < rootV) {  
 return false;  
 }  
 }  
  
 return isBST(sequence, start, rightIndex - 1) && isBST(sequence, rightIndex, end - 1);  
}