

题目 二叉搜索树的后序遍历序列

考点 举例让抽象具体化 热点指数 54280 通过率 23.53%

具体题目

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。

采用分治法思想，找到根结点、左子树的序列、右子树的序列，分别判断左右子序列是否为二叉树的后序序列。由题意可得：

1. 后序遍历序列的最后一个元素为二叉树的根节点；
2. 二叉搜索树左子树上所有的结点均小于根结点、右子树所有的结点均大于根结点。

算法步骤如下：

- 找到根结点；
- 遍历序列，找到第一个大于等于根结点的元素*i*，则*i*左侧为左子树、*i*右侧为右子树；
- 我们已经知道*i*左侧所有元素均小于根结点，那么再依次遍历右侧，看是否所有元素均大于根结点；若出现小于根结点的元素，则直接返回false；若右侧全都大于根结点，则：
- 分别递归判断左/右子序列是否为后序序列；

```
public boolean VerifySequenceOfBST(int [] sequence) {
    if ( sequence==null || sequence.length<=0 ) return false;
    return VerifySequenceOfBST(sequence, 0, sequence.length-1);
}
private boolean VerifySequenceOfBST(int [] sequence, int start, int end){
    if ( start>=end )
        return true;

    int root = sequence[end];

    int i=start;
    while( sequence[i] < root ){
        i++;
    }

    int j=i;
    while( j<end ){
        if ( sequence[j]<root ) {
            return false;
        }
        j++;
    }

    boolean left = VerifySequenceOfBST(sequence, start, i-1);
    boolean right = VerifySequenceOfBST(sequence, i, end-1);
    return left && right;
}
```