

剑指Offer（二十三）： 二叉搜索树的后序遍历序列

© 2017年12月12日 10:32:39 1 5,613 °C 编辑



一、前言

本系列文章为《剑指Offer》刷题笔记。

刷题平台：[牛客网](#)

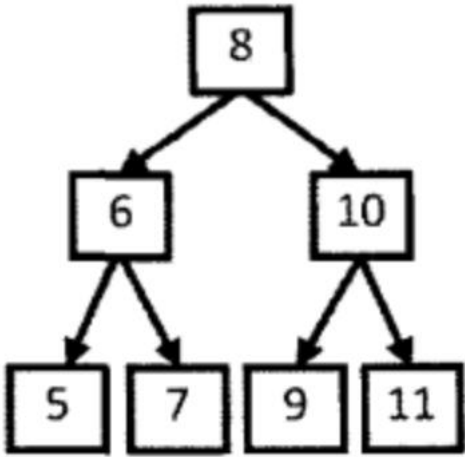
书籍下载：[共享资源](#)

二、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。

1、思路

举例说明：



以{5,7,6,9,11,10,8}为例，后序遍历结果的最后一个数字8就是根结点的值。在这个数组中，前3个数字5、7和6都比8小，是值为8的结点的左子树结点；后3个数字9、11和10都比8大，是值为8的结点的右子树结点。

我们接下来用同样的方法确定与数组每一部分对应的子树的结构。这其实就是一个递归的过程。对于序列5、7、6，最后一个数字6是左子树的根结点。数字5比6小，是值为6的结点的左子结点，而7则是它的右子结点。同样，在序列9、11、10中，最后一个数字10是右子树的根结点，数字9比10小，是值为10的结点的左子结点，而11则是它的右子结点。

我们使用递归的方法，先判断数组的左子树和右子树的位置，然后再判断左子树、右子树是不是二叉搜索树。

2、代码

C++：

```
1 class Solution {
2 public:
3     bool VerifySequenceOfBST(vector<int> sequence) {
4         return bst(sequence, 0, sequence.size() - 1);
5     }
6 private:
7     bool bst(vector<int> seq, int begin, int end){
8         if(seq.empty() || begin > end){
9             return false;
10        }
11
12        //根结点
13        int root = seq[end];
14
15        //在二叉搜索树中左子树的结点小于根结点
16        int i = begin;
17        for(; i < end; ++i){
18            if(seq[i] > root){
19                break;
20            }
21        }
22
23        //在二叉搜索书中右子树的结点大于根结点
24        for(int j = i; j < end; ++j){
25            if(seq[j] < root){
26                return false;
27            }
28        }
29
30        //判断左子树是不是二叉搜索树
31        bool left = true;
32        if(i > begin){
33            left = bst(seq, begin, i - 1);
34        }
35
36        //判断右子树是不是二叉搜索树
37        bool right = true;
38        if(i < end - 1){
39            right = bst(seq, i, end - 1);
40        }
41
42        return left && right;
43    }
44 };
```

#### Python:

```
1 # -*- coding:utf-8 -*-
2 class Solution:
3     def VerifySequenceOfBST(self, sequence):
4         # write code here
5         if not len(sequence):
6             return False
7         if len(sequence) == 1:
8             return True
9         length = len(sequence)
10        root = sequence[-1]
11        i = 0
12        while sequence[i] < root:
13            i = i + 1
14        k = i
15        for j in range(i, length-1):
16            if sequence[j] < root:
17                return False
18        left_s = sequence[:k]
19        right_s = sequence[k:length-1]
20        left, right = True, True
21        if len(left_s) > 0:
22            left = self.VerifySequenceOfBST(left_s)
23        if len(right_s) > 0:
24            right = self.VerifySequenceOfBST(right_s)
25        return left and right
```

Pytho



微信公众号

分享技术，乐享生活：微信公众号搜索

「JackCui-AI」关注一个在互联网摸爬滚  
打的潜行者。

简洁是智慧的灵魂，冗长是肤浅的藻饰。--- 莎士比亚