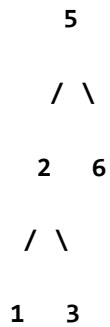


二叉搜索树的后序遍历序列

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结果。如果是则返回 `true`，否则返回 `false`。假设输入的数组的任意两个数字都互不相同。

参考以下这颗二叉搜索树：



示例 1：

输入：[1,6,3,2,5]

输出：false

示例 2：

输入：[1,3,2,6,5]

输出：true

解题思路

1. 最后一个节点为根节点
2. 左边的节点全部要小于根，右边的节点全部要大于根，因此数组可以分成两个区间，前半部分全部小于根，后半部分全部大于根
3. 找到两个区间的分割点，判断是否两个区间是否符合

```
class Solution {
public:
    bool dfs(vector<int>& postorder, int start, int end)
    {
        if(start >= end) return true;
        int postroot = postorder[end];
        int div = start;
        while(postorder[div] < postroot) div++; // 根据左子树小于根 右子树大于根的关系 找到左右子树分界点
        for(int i = div; i < end; i++)
        {
            if(postorder[i] < postroot) return false; // 违反右子树大于根的关系
        }
    }
};
```

```
        return dfs(postorder,start,div-1)&&dfs(postorder,div,end-1);
    }
    bool verifyPostorder(vector<int>& postorder) {
        if(postorder.size()<2) return true;
        return dfs(postorder,0,postorder.size()-1);
    }
};
```