# 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：

5

/ \

2 6

/ \

1 3

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

提示：

数组长度<= 1000

# 分析

**思路：**

根据二叉树的定义来看最后一个是根节点前面的序列中必须存在一个拐点拐点前后的值要么均大于根要么均小于根。

那么可以将序列与根节点比较，递归逐步缩小序列范围，注意边界条件比如全部比根节点大或者全部比根节点小。

最后设置退出条件。

**代码：**

class Solution {

public:

vector<int> res;

bool verifyPostorder(vector<int>& postorder) {

res = postorder;

return dfs(0, postorder.size() - 1);

}

bool dfs(int l, int r)

{

if(l >= r) return true;//退出条件

int root = res[r];//最后一个点是根结点

int k = l;//从最左边开始

while(k < r && res[k] < root) k++;//符合左子树的点

for(int i = k; i < r; i++)//此时的k是右子树的第一个点

{

if(res[i] < root)//如果右子树小于根，说明不符合

return false;

}

return dfs(l, k - 1) && dfs(k, r - 1);//逐步缩小范围

}

};