## 题目描述：

## 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：

BST的后序序列的合法序列是，对于一个序列S，最后一个元素是x （也就是根），如果去掉最后一个元素的序列为T，那么T满足：T可以分成两段，前一段（左子树）小于x，后一段（右子树）大于x，且这两段（子树）都是合法的后序序列。然后通过递归判断左右子树是否为BST；

关键就是找到左子树的根节点，然后递归判断

## 算法描述：

非递归：

class Solution {

public:

bool VerifySquenceOfBST(vector<int> sequence) {

int size = sequence.size();

if(size == 0)

return false;

int i=0;

size = size -1;

while(size >=0)

{

while(sequence[i] < sequence[size])

{

i++;

}

while(sequence[i] > sequence[size])

{

i++;

}

if(i < size )

return false;

i = 0;

size--;

}

return true;

}

};

递归：

class Solution {

bool judge (vector<int> a, int begin, int end)

{

if (begin >= end )

return true;

int i=end-1;

while (i >= begin && a[i] > a[end])

i--;//i为左子树的根节点

for (int j=i; j>=begin;j--)

if (a[j] > a[end])

return false;

return judge(a,begin,i)&&judge(a,i+1,end-1);

}

public:

bool VerifySquenceOfBST(vector<int> sequence) {

if (!sequence.size())

return false;

return judge(sequence,0,sequence.size()-1);

}

};