# 31. 栈的压入、弹出序列

## 题目链接

[牛客网](https://www.nowcoder.com/practice/d77d11405cc7470d82554cb392585106?tpId=13&tqId=11174&tPage=1&rp=1&ru=/ta/coding-interviews&qru=/ta/coding-interviews/question-ranking&from=cyc_github)

## 题目描述

输入两个整数序列，第一个序列表示栈的压入顺序，请判断第二个序列是否为该栈的弹出顺序。假设压入栈的所有数字均不相等。

例如序列 1,2,3,4,5 是某栈的压入顺序，序列 4,5,3,2,1 是该压栈序列对应的一个弹出序列，但 4,3,5,1,2 就不可能是该压栈序列的弹出序列。

## 解题思路

使用一个栈来模拟压入弹出操作。每次入栈一个元素后，都要判断一下栈顶元素是不是当前出栈序列 popSequence 的第一个元素，如果是的话则执行出栈操作并将 popSequence 往后移一位，继续进行判断。

public boolean IsPopOrder(int[] pushSequence, int[] popSequence) {
 int n = pushSequence.length;
 Stack<Integer> stack = new Stack<>();
 for (int pushIndex = 0, popIndex = 0; pushIndex < n; pushIndex++) {
 stack.push(pushSequence[pushIndex]);
 while (popIndex < n && !stack.isEmpty()
 && stack.peek() == popSequence[popIndex]) {
 stack.pop();
 popIndex++;
 }
 }
 return stack.isEmpty();
}