



链滴

栈的压入、弹出序列

作者: [yudake](#)

原文链接: <https://ld246.com/article/1518836853653>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

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是该栈的弹出顺序。假设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。（注意：这两个序列的长度是相等的）

解题思路

- 将pushA依次压入栈；
- 对比栈顶元素和popA[j]元素是否相等：
 - 如果相等，弹出栈顶；
 - 如果不等，继续压栈。

如果栈为空则返回true，否则返回false。

```
import java.util.Stack;
```

```
public class Solution {  
    public boolean IsPopOrder(int [] pushA,int [] popA) {  
        if (pushA == null || pushA.length == 0)  
            return false;  
        Stack<Integer> stack = new Stack<>();  
        int j = 0;  
        for (int i = 0; i < pushA.length; i++) {  
            stack.push(pushA[i]);  
            while (j < popA.length && stack.peek() == popA[j]) {  
                stack.pop();  
                j++;  
            }  
        }  
        return stack.isEmpty();  
    }  
}
```