# 栈的压入、弹出序列

## 题目

[牛客网](https://www.nowcoder.com/practice/d77d11405cc7470d82554cb392585106?tpId=13&tqId=11174&rp=1&ru=%2Fta%2Fcoding-interviews&qru=%2Fta%2Fcoding-interviews%2Fquestion-ranking&tPage=1)

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可能为该栈的弹出顺序。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。（注意：这两个序列的长度是相等的）

## 解题思路

1. 通过 Stack 进行模拟 push，当 pop 的节点等于 Stack 的 top 节点时，pop Stack
2. 最后如果 Stack 剩余数据，则判定为 false

public boolean IsPopOrder(int[] pushA, int[] popA) {
 if (pushA.length != popA.length) {
 return false;
 }

 if (pushA.length == 0) {
 return false;
 }

 LinkedList<Integer> stack = new LinkedList<>();

 int j = 0;
 for (int value : pushA) {
 stack.addLast(value);

 while (stack.peekLast() != null && popA[j] == stack.getLast()) {
 j++;
 stack.removeLast();
 }
 }

 return stack.isEmpty();
}