

剑指Offer（二十一）：栈的压入、弹出序列

🕒 2017年12月11日 10:13:29 🗨 7 🌡 3,439 °C 📝 编辑



一、前言

本系列文章为《剑指Offer》刷题笔记。

刷题平台：[牛客网](#)

书籍下载：[共享资源](#)

二、题目

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。（注意：这两个序列的长度是相等的）

1、思路

借用一个辅助的栈，遍历压栈顺序，先讲第一个放入栈中，这里是1，然后判断栈顶元素是不是出栈顺序的第一个元素，这里是4，很显然1≠4，所以我们继续压栈，直到相等以后开始出栈，出栈一个元素，则将出栈顺序向后移动一位，直到不相等，这样循环等压栈顺序遍历完成，如果辅助栈还不为空，说明弹出序列不是该栈的弹出顺序。

2、代码


C++:

```
1 class Solution {
2 public:
3     bool IsPopOrder(vector<int> pushV,vector<int> popV) {
4         if(pushV.size() == 0){
5             return false;
6         }
7         for(int i = 0, j = 0; i < pushV.size();i++){
8             stackData.push(pushV[i]);
9             while(j < popV.size() && stackData.top() == popV[j]){
10                 stackData.pop();
11                 j++;
12             }
13         }
14         return stackData.empty();
15     }
16 private:
17     stack<int> stackData;
18 };
```

Python2.7:

```
1 # -*- coding:utf-8 -*-
2 class Solution:
3     def IsPopOrder(self, pushV, popV):
4         # write code here
5         if len(popV) == 0 or len(pushV) != len(popV):
6             return False
7         stackData = []
8         for i in pushV:
9             stackData.append(i)
10            while len(stackData) and stackData[-1] == popV[0]:
11                stackData.pop()
12                popV.pop(0)
13            if len(stackData):
```

```
14         return False
15     return True
```



微信公众号

分享技术，乐享生活：微信公众号搜索「JackCui-AI」关注一个在互联网摸爬滚打的潜行者。

stay hungry , stay foolish.--- Steve Jobs