

P4387 【深基15. 习9】验证栈序列

题目描述

给出两个序列 `pushed` 和 `poped` 两个序列，其取值从 1 到  $n$  ( $n \leq 100000$ )。已知入栈序列是 `pushed`，如果出栈序列有可能是 `poped`，则输出 `Yes`，否则输出 `No`。为了防止骗分，每个测试点有多组数据。

输入格式

第一行一个整数  $q$ ，询问次数。

接下来  $q$  个询问，对于每个询问：

第一行一个整数  $n$  表示序列长度；

第二行  $n$  个整数表示入栈序列；

第二行  $n$  个整数表示出栈序列；

输出格式

对于每个询问输出答案。

输入样例

```
2
5
1 2 3 4 5
5 4 3 2 1
4
1 2 3 4
2 4 1 3
```

输出样例

```
Yes
No
```

解析

依然是一道阅读理解题，首先这道题绝对不是让你比较两个序列颠倒之后是否相同！

对于入栈序列，它可以有多种多样的出栈顺序，而我们就是要在它众多的顺序中判断序列2是否使其中之一。

举例，对于栈1 2 3 4 5，它的出栈顺序可能有：

a、1 2 3 4 5，即入1个马上就出一个。

b、2、1、3、4、5，即入到2，然后全部出栈，接着入1个出1个。

题意搞懂了，那么怎么判断出栈序列是否是入栈序列众多出栈方式之一呢？很简单，我们将序列1进行入栈，当它的栈顶与出栈序列相同时，就执行出栈。如果入栈序列最后能够全部出栈，则表示两个序列是匹配的。

## 编码

```
#include<bits/stdc++.h>

using namespace std;
stack<int> q; //栈q
vector<int> a, b;
int p, n; //p组数据，n为序列长度
int main() {
    cin >> p;
    while (p--) {
        cin >> n;
        int sum = 0; //入栈队列a，待检验队列b，计数器sum
        //读入数据
        for (int i = 0; i < n; i++) {
            int t;
            cin >> t;
            a.push_back(t);
        }
        for (int i = 0; i < n; i++) {
            int t;
            cin >> t;
            b.push_back(t);
        }
        for (int i = 0; i < n; i++) {
            q.push(a[i]); //入栈
            //当栈顶元素与b中当前元素相同时出栈
            while (!q.empty() && (q.top()) == b[sum]) {
                //尝试出栈
                q.pop();
                //sum++到b下一个元素
                sum++;
            }
        }
    }
}
```

```
    //如果栈为空说明出栈序列b正确
    if (q.empty()) {
        cout << "Yes" << endl;
    } else {
        cout << "No" << endl;
    }
    //清空数据，为下一次做准备
    while (!q.empty()) {
        q.pop();
    }
    a.clear();
    b.clear();
}
return 0;
}
```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

