Problem 946: Validate Stack Sequences

Problem Information

Difficulty: Medium
Acceptance Rate: 69.95%
Paid Only: No
Tags: Array, Stack, Simulation

Problem Description

Given two integer arrays `pushed` and `popped` each with distinct values, return `true` _if this could have been the result of a sequence of push and pop operations on an initially empty stack, or_`false` _otherwise._

Example 1:

Input: pushed = [1,2,3,4,5], popped = [4,5,3,2,1] **Output:** true **Explanation:** We might do the following sequence: push(1), push(2), push(3), push(4), pop() -> 4, push(5), pop() -> 5, pop() -> 3, pop() -> 2, pop() -> 1

Example 2:

Input: pushed = [1,2,3,4,5], popped = [4,3,5,1,2] **Output:** false **Explanation:** 1 cannot be popped before 2.

Constraints:

* `1 <= pushed.length <= 1000` * `0 <= pushed[i] <= 1000` * All the elements of `pushed` are **unique**. * `popped.length == pushed.length` * `popped` is a permutation of `pushed`.

Code Snippets

C++:

```
class Solution {
public:
bool validateStackSequences(vector<int>& pushed, vector<int>& popped) {

}
};
```

Java:

```
class Solution {
public boolean validateStackSequences(int[] pushed, int[] popped) {

}
}
```

Python3:

```
class Solution:
def validateStackSequences(self, pushed: List[int], popped: List[int]) ->
bool:
```