946. Validate Stack Sequences//Stack Permutations (Check if an array is stack permutation of other)

Given two sequences `pushed` and `popped` **with distinct values**, return `true` if and only if this could have been the result of a sequence of push and pop operations on an initially empty stack.

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:

- `0 <= pushed.length == popped.length <= 1000`
- `0 <= pushed[i], popped[i] < 1000`
- `pushed` is a permutation of `popped`.
- `pushed` and `popped` have distinct values.

```
def validateStackSequences(self, pushed: List[int], popped: List[int]) -> bool:
        n = len(pushed)
        stack = []
        j = 0
        for x in pushed:
            stack.append(x)
            while stack and j<len(popped) and stack[-1]==popped[j]:
                stack.pop()
                j = j+1
        return j==len(popped)
```