

Validate Subsequence

array = [5, 1, 22, 25, 6, -1, 8, 10]
sequence = [1, 6, -1, 10]
true

input: Two non-empty arrays of integers

output: return true if second array is a subsequence of the first one
return false otherwise

Note: A subsequence of an array is the set of numbers that aren't necessarily adjacent in the array but that are in the same order as they appear in the array.

$O(n)$ time ; $O(1)$ space

```
// O(n) time | O(1) space
function validateSequence(arr, seq) {
  let arrIndex = 0;
  let seqIndex = 0;
  while (arrIndex < arr.length && seqIndex < seq.length) {
    if (arr[arrIndex] === seq[seqIndex]) {
      seqIndex++;
    }
    arrIndex++;
  }
  return seqIndex === seq.length;
}
```

array = [5, 1, 22, 25, 6, -1, 8, 10]
sequence = [1, 6, -1, 10]

We start with two pointers that point at the starting index of the array and the sequence.

We iterate through the array. At each iteration, we check if the value at starting array index is in the current index of the sequence.

If the value is in the sequence, we advance the sequence pointer.

When done traversing the array we check if the sequence index is equal to the sequence array length. This tells us the sequence array is a subsequence of array.

- $O(n)$ time since we are traversing the array
- $O(1)$ space since we are not using more space as the input grows