SWIFT RAMEN

AUG 3

Validate array subsequence

ALGORITHMS (/HOME/CATEGORY/ALGORITHMS), INTERVIEW CHALLENGE (/HOME/CATEGORY/INTERVIEW+CHALLENGE)

Problem statement

Write a function that determines whether an array is a subsequence of another array. A subsequence is a set of numbers that are not necessarily adjacent to each other in the second array but appear in the same order as in the first array. For example, the array `[3, 5, 2]` is valid a subsequence of the array `[1, 3, 4, 5, 6, 2]`.

Sample input

```
array = [8, 3, 23, 27, 5, 0, 9, 11]
sequence = [3, 5, 0, 11]
```

Sample output

```
true
```

The concept

You can use a counter to check that each successive element of the sequence array comes one after the other in the first array. You can then iterate over the entirety of the first array to check if the element in the sequence array exists in the first array. By the end of the loop if the value of the counter is equal to the count of the sequence index, it means that each elements in the sequence index was found in the same order in the first array and you will get `true`. Let's code this out now.

Solution

Add the code below to your playground file.

```
import UIKit

func isValidSubsequence(array: [Int], sequence: [Int]) -> Bool {
  /// 1
  var sequenceIndex = 0
  /// 2
  for value in array {
    /// 3
    if sequenceIndex == sequence.count { break }
    /// 4
    if value == sequence[sequenceIndex] { sequenceIndex += 1 }
  }
  /// 5
  return sequenceIndex == sequence.count
}
```

Here's what's happening:

1. You maintain an index property for the **sequence** array.
2. You loop over each element of **array**
3. You check if the **sequenceIndex** is equal to the count of the **sequence** array. If this check is `true` then you break out of the loop. This is optimization helps us not loop over the entire array every time, if our solution condition is satisfied then we break out of the loop and hit the `return` statement.
4. You check if an element from the **sequence** array at index **sequenceIndex** is equal to the current element **value** from **array**. If it evaluates to `true` you update the

SWIFT RAMEN

sequenceIndex by incrementing it by **1**.

5. You return the result of the conditional check, whether the value of **sequenceIndex** is equal to the count of the array **sequence**.

Test

Add the code below to test out the implementation

```
/// Test
let array = [8, 3, 23, 27, 5, 0, 9, 11]
let sequence = [3, 5, 0, 11]
isValidSubsequence(array: array, sequence: sequence)
```

You will see the result in the inspector on the right..

```
true
```

Success!

Complexity

Space

Since you are not storing anything except the **sequenceIndex** which will always have a small finite value, the space complexity is **O(1)**.

Time

Even though we optimize our for loop by breaking out of it if our solution condition is met, we consider the worst case scenario, where the loop iterates over every element in the array. This gives us a time complexity of **O(n)** ,where **n** is the number of elements in the array.

SWIFT RAMEN

If you have any suggestions or corrections to suggest in the content of this article please let me know on twitter @azharcodes (https://twitter.com/azharcodes). Cheers!

AZHAR ANWAR (/?AUTHOR=5CA7B7D96F935E582E4C8062)

algorithms (/home/tag/algorithms), swift algorithms (/home/tag/swift+algorithms), ios interview (/home/tag/ios+interview), swift interview (/home/tag/swift+interview)

(HTTPS://TWITTER.COM/INTENT/TWEET?URL=HTTPS%3A%2F%2FWWW.SWIFTRAMEN.COM%2FHOME%2FVALIDATE-ARRAY-SUBSEQUENCE&TEXT=WRITE+A+FUNCTION+THAT+DETERM

(HTTPS://WWW.LINKEDIN.COM/SHAREARTICLE?MINI=TRUE&SOURCE=SWIFT+RAMEN&SUMMARY=WRITE+A+FUNCTION+THAT+DETERMINES+WHETHER+AN+ARRAY+IS+A+SUBSEQUEN

(HTTPS://WWW.REDDIT.COM/SUBMIT?URL=HTTPS%3A%2F%2FWWW.SWIFTRAMEN.COM%2FHOME%2FVALIDATE-ARRAY-SUBSEQUENCE)

Leave a comment

‹

Aug 4 Longest peak in array

(/home/longest-peak-in-array)

›

Jul 30 Closure initialization

(/home/closure-initialization)

SWIFT RAMEN