Using an Array as a built in index for LeetCode problems

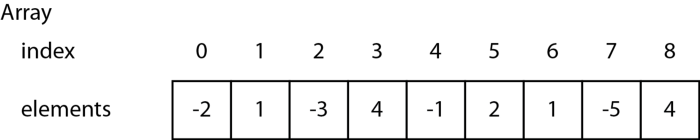
Algorithms in Swift

[](https://medium.com/@stevenpcurtis.sc?source=post_page-----1787acfb0f4e----------------------)

[Steven Curtis](https://medium.com/@stevenpcurtis.sc?source=post_page-----1787acfb0f4e----------------------)

[Jun 26](https://medium.com/@stevenpcurtis.sc/using-an-array-as-an-built-index-for-leetcode-problems-1787acfb0f4e?source=post_page-----1787acfb0f4e----------------------) · 4 min read





An example array with index places shown

**Before We Start**

**Why?**

O(n) solutions are often required in Interviews, and for some LeetCode problems. Rather than using extra space, we can use the property of arrays that they are ordered and have an intrinsic index in order to (often) find a missing Integer in an unsorted list

**Prerequisites**

To follow along you should be able to use Arrays, and understand the fact that they are zero-indexed. Some experience of using inout parameters could be beneficial (or you can just go with it).

**Keywords and Terminology**

* Array: An ordered, random-access collection

**Setting elements to their correct position**

Let us make the example as simple as possible, and use only positive integers in an array. The challenge is to put them into a the correct order in an array

**The sort solution:**

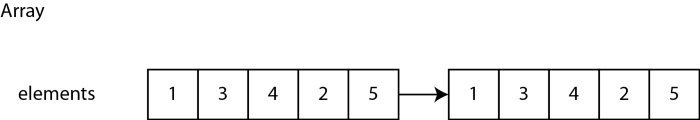
This won’t get you far in an interview, but you might choose to use a simple in-built sort in Swift:

*arr.sort()*

**A real in-place solution:**

This involves quite a bit more thought. We are going to transform the unordered array [1,3,4,2,5] to [1,2,3,4,5]

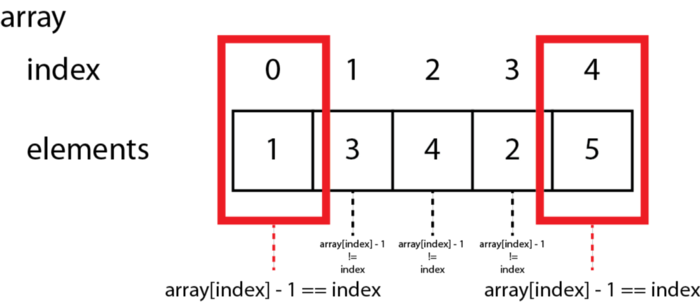




The target transformation for an in place sort using the natural index in an array

Restoring the index to our target array we intuitively know that the first element is in the “right” position (as is the last element):





Finding the two “out of place” elements using the formula array[index] — 1 == index

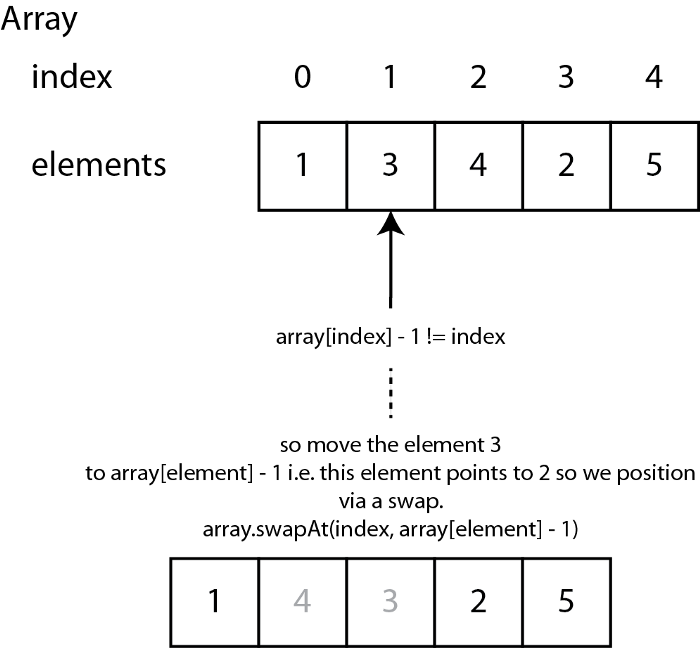
This is because **array[index] — 1 == index**.We note that it is we minus 1 from where the element “points to” as the index of the array is zero-indexed

We will move through each element in the array, and if array[index] != index -1 we will move the element into the right place in the array.

**The first index of an out-of-place element**

at index 1, the element 3 is out of place, so is swapped with the element that \*should\* be in it’s place.

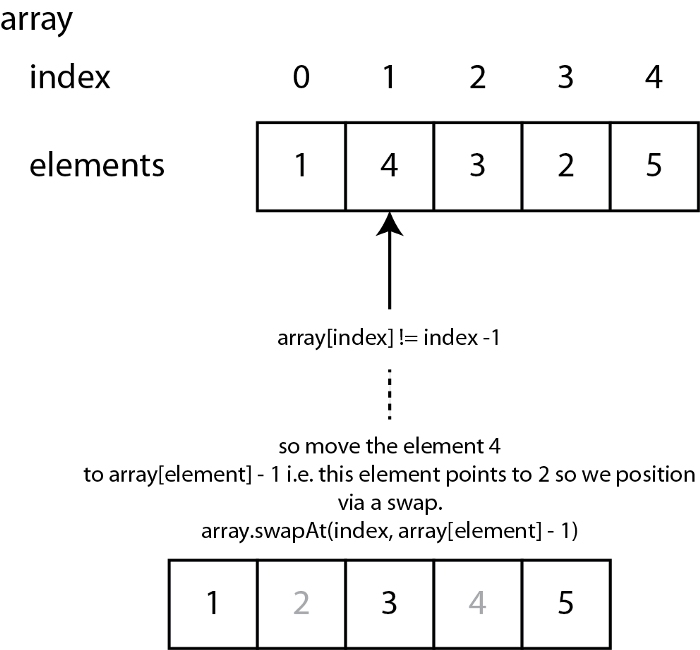




it! at index we still don’t have the right element at the position!

We have to **keep going** at the same index, until it has the right element at that





This is resolved through a simple while loop in Swift:

|  |  |
| --- | --- |
|  | for index in 0..<array.count { |
|  | while array[index] - 1 != index { |
|  | array.swapAt(index, array[index] - 1) |
|  | } |
|  | } |

[**view raw**](https://gist.github.com/stevencurtis/1245cb51398b8c869858aac38e52ec9e/raw/c7eaa0c24297cf87145c67927d68a1a6ae7745cf/inplacearraywhileloop)

[**inplacearraywhileloop**](https://gist.github.com/stevencurtis/1245cb51398b8c869858aac38e52ec9e#file-inplacearraywhileloop) hosted with ❤ by [**GitHub**](https://github.com/)

and in our example the final iterations will find the elements all in the right place, and as a result we have a sorted array.

**Extending for LeetCode problems**

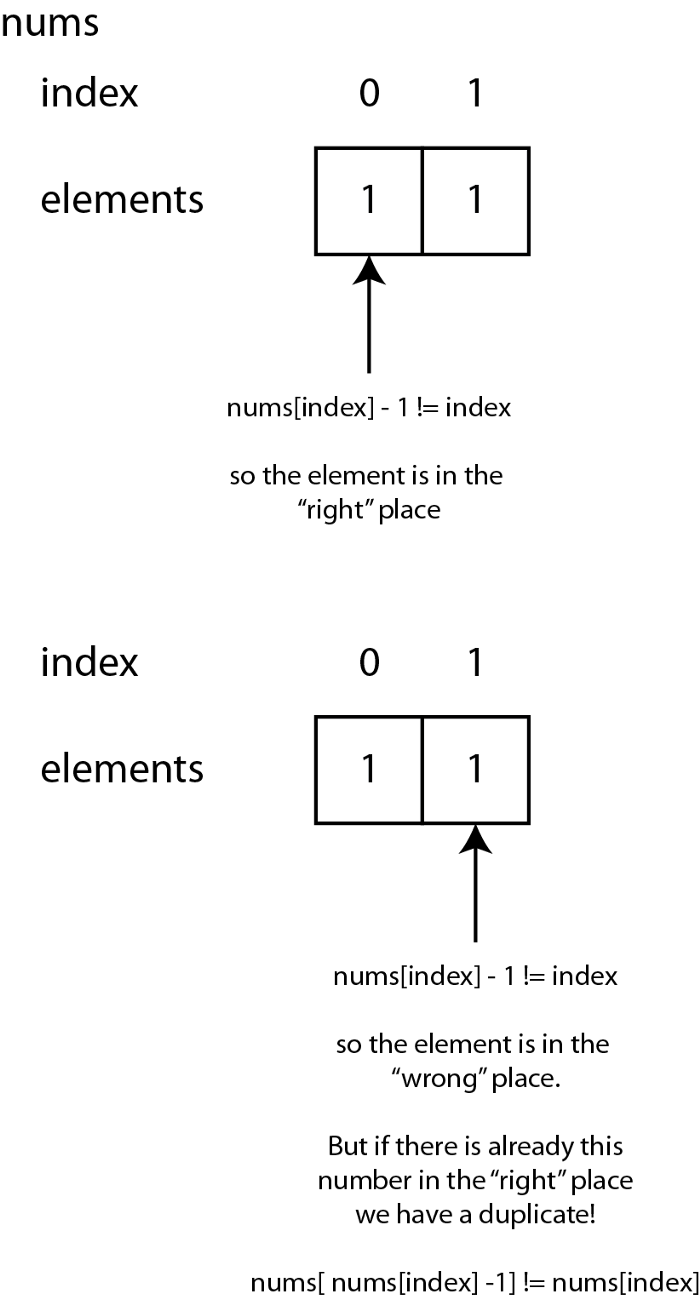
An example of where this code is used in LeetCode is 287. Find the Duplicate Number which expects you to identify the duplicate number, where each element Integer is between 1 and n (inclusive).

Since there is only one duplicate number (in the problem description), we can realise that it is possible to exit early.

*How do we find that there is a duplicate number?*

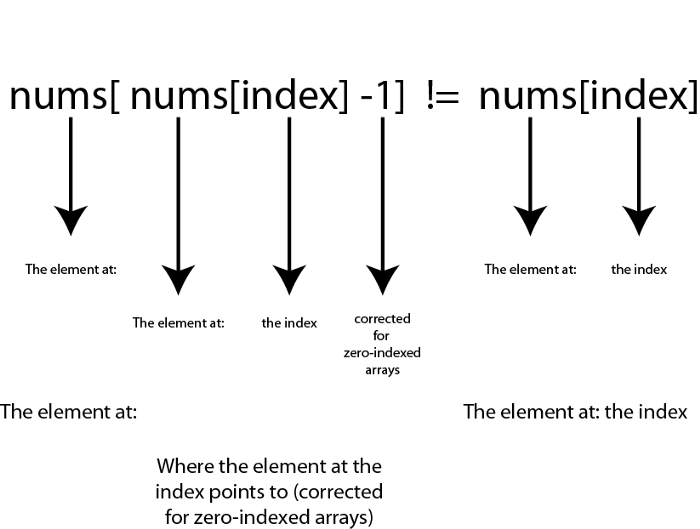
Let us step through an example:





This last expression might need some explaination.





|  |  |
| --- | --- |
|  | func findDuplicate(\_ nums: [Int]) -> Int { |
|  | var nums = nums |
|  | for i in 0..<nums.count { |
|  | while nums[i] - 1 != (i) { |
|  | if nums[ nums[i] - 1] == nums[i] { |
|  | return nums[i] |
|  | } |
|  | nums.swapAt(i, nums[i] - 1) |
|  | } |
|  | } |
|  | return 0 |
|  | } |

[**view raw**](https://gist.github.com/stevencurtis/ff2a1f1aa279aaae17f81400ff8667f5/raw/47865074867db9f80a397510afca08536f4f9274/findduplicatesolution)

[**findduplicatesolution**](https://gist.github.com/stevencurtis/ff2a1f1aa279aaae17f81400ff8667f5#file-findduplicatesolution) hosted with ❤ by [**GitHub**](https://github.com/)

Giving us the following Swift code:

|  |  |
| --- | --- |
|  | func findDuplicate(\_ nums: [Int]) -> Int { |
|  | var nums = nums |
|  | for i in 0..<nums.count { |
|  | while nums[i] - 1 != (i) { |
|  | if nums[ nums[i] - 1] == nums[i] { |
|  | return nums[i] |
|  | } |
|  | nums.swapAt(i, nums[i] - 1) |
|  | } |
|  | } |
|  | return 0 |
|  | } |

[**view raw**](https://gist.github.com/stevencurtis/ff2a1f1aa279aaae17f81400ff8667f5/raw/47865074867db9f80a397510afca08536f4f9274/findduplicatesolution)

[**findduplicatesolution**](https://gist.github.com/stevencurtis/ff2a1f1aa279aaae17f81400ff8667f5#file-findduplicatesolution) hosted with ❤ by [**GitHub**](https://github.com/)

**Conclusion**

This is a common template for problems in LeetCode

I hope you enjoyed this tutorial. You can find the code and repo below. Thanks for reading!

**The GitHub link:**

**[stevencurtis/IndexForArray](https://github.com/stevencurtis/IndexForArray?source=post_page-----1787acfb0f4e----------------------)**

[A generalised solution for a Medium post. Contribute to stevencurtis/IndexForArray development by creating an account…](https://github.com/stevencurtis/IndexForArray?source=post_page-----1787acfb0f4e----------------------)

[github.com](https://github.com/stevencurtis/IndexForArray?source=post_page-----1787acfb0f4e----------------------)

**Want to get in contact? Get in touch on Twitter:**

**[Steven Curtis (@stevenpcurtis) | Twitter](https://twitter.com/stevenpcurtis?source=post_page-----1787acfb0f4e----------------------)**

[The latest Tweets from Steven Curtis (@stevenpcurtis). Studying for a Masters in Computing while developing iOS Apps…](https://twitter.com/stevenpcurtis?source=post_page-----1787acfb0f4e----------------------)

[twitter.com](https://twitter.com/stevenpcurtis?source=post_page-----1787acfb0f4e----------------------)

1

* [Programming](https://medium.com/tag/programming) [Swift](https://medium.com/tag/swift) [Leetcode](https://medium.com/tag/leetcode)