Interview Cake

In-Place Algorithm

An **in-place** algorithm operates *directly* on its input and *changes* it, instead of creating and returning a *new* object. This is sometimes called **destructive**, since the original input is "destroyed" when it's edited to create the new output.

Careful: "In-place" does *not* mean "without creating any additional variables"! Rather, it means "without creating a new copy of the input." In general, an in-place function will only create additional variables that are $O(1)$ space.

Here are two functions that do the same operation, except one is in-place and the other is out-of-place:

JavaScript ▾

```
function squareArrayInPlace(intArray) {

    intArray.forEach(function(int, index) {
        intArray[index] *= int;
    });

    // NOTE: we don't *need* to return anything
    // this is just a convenience
    return intArray;
}

function squareArrayOutOfPlace(intArray) {

    // we allocate a new array with the length of the input array
    var squaredArray = [];

    intArray.forEach(function(int, index) {
        squaredArray[index] = Math.pow(int, 2);
    });

    return squaredArray;
}
```

Working in-place is a good way to save space. An in-place algorithm will generally have $O(1)$ space cost.

But be careful: an in-place algorithm can cause side effects. Your input is "destroyed" or "altered," which can affect code *outside* of your function. For example:

JavaScript ▾

```
var originalArray = [2, 3, 4, 5];
var squaredArray  = squareArrayInPlace(originalArray);

console.log('squared: ' + squaredArray);
// logs: squared: 4,9,16,25

System.out.println("original array: " + originalArray);
// logs: original array: 4,9,16,25 - confusingly!

// and if squareArrayInPlace() didn't return anything,
// which it could reasonably do, squaredArray would be undefined!
```

Generally, out-of-place algorithms are considered safer because they avoid side effects. You should only use an in-place algorithm if you're very space constrained or you're *positive* you don't need the original input anymore, even for debugging.

In-Place Algorithm Coding Interview Questions

24 **Reverse A Linked List »**

Write a function to reverse a linked list in-place. keep reading »

(/question/reverse-linked-list)

26✔ **Reverse String in Place »**

Write a function to reverse a string in-place. keep reading »

(/question/reverse-string-in-place)

27✔ **Reverse Words »**

Write a function to reverse the word order of a string, in-place. It's to decipher a supersecret message and win the war. keep reading »

(/question/reverse-words)

40 **Find Repeat, Space Edition »**

Figure out which number is repeated. But here's the catch: optimize for space. keep reading »

(/question/find-duplicate-optimize-for-space)

All Questions ➡ (/all-questions)