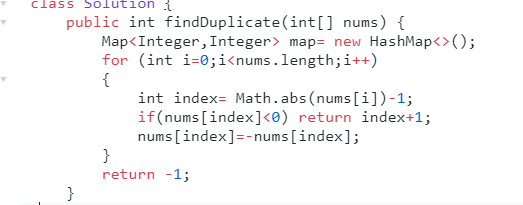
# Arrays:

1. **Find the Duplicate Number in the array.**

You must use only constant, *O*(1) extra space. Your runtime complexity should be less than *O*(*n*2).

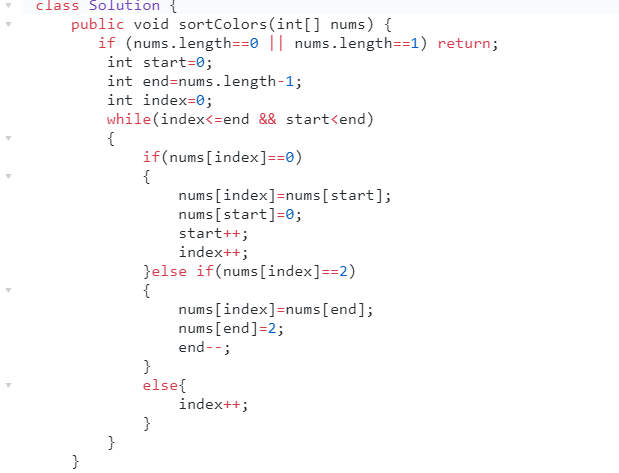
Code: ( <https://www.youtube.com/watch?v=aMsSF1Il3IY> )



Also see “Floyd’s Tortoise-Hare” cycle detection technique. (<https://leetcode.com/problems/find-the-duplicate-number/solution/>)

1. **Sort an array of 0’s 1’s 2’s without using extra space or sorting algo:**

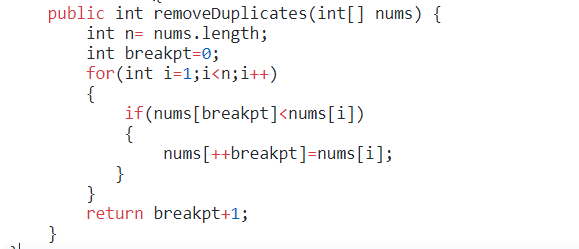
**(**<https://leetcode.com/problems/sort-colors/>**)**



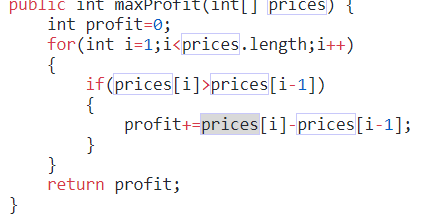
<https://www.youtube.com/watch?v=uvB-Ns_TVis>

1. Given a sorted array nums, remove the duplicates [**in-place**](https://en.wikipedia.org/wiki/In-place_algorithm) such that each element appear only once and return the new length. Do not allocate extra space for another array, you must do this by **modifying the input array**[**in-place**](https://en.wikipedia.org/wiki/In-place_algorithm) with O(1) extra memory.

<https://leetcode.com/explore/interview/card/top-interview-questions-easy/92/array/727/>

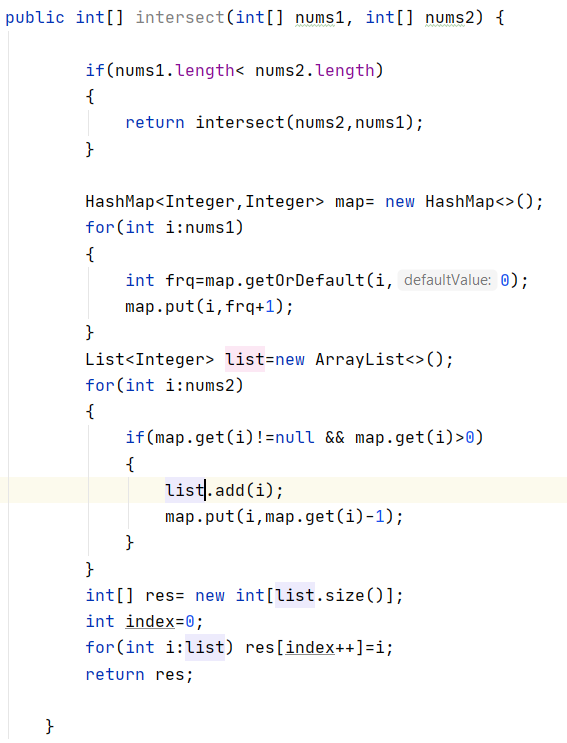


1. **Best Time to Buy and Sell Stock II**



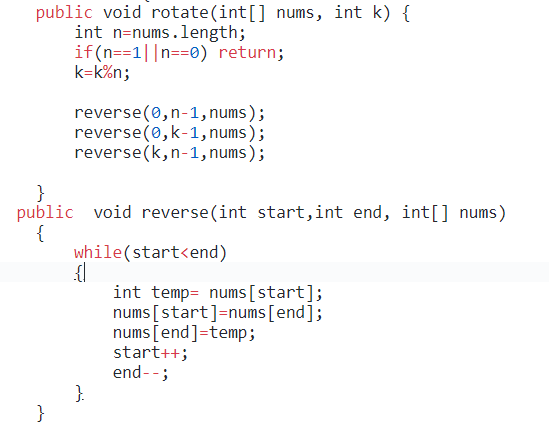
1. **Given two arrays, write a function to compute their intersection.**

Each element in the result should appear as many times as it shows in both arrays. The result can be in any order.



1. **Given an array, rotate the array to the right by k steps, where k is non-negative.**

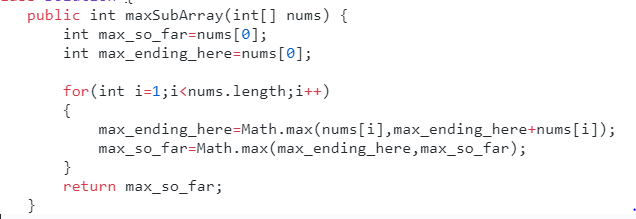
<https://leetcode.com/explore/interview/card/top-interview-questions-easy/92/array/646/>



1. Given an integer array nums, find the contiguous subarray (containing at least one number) which has the largest sum and return its sum. (Kadane's Algorithm)

<https://www.youtube.com/watch?v=2MmGzdiKR9Y&t=746s>

<https://leetcode.com/problems/maximum-subarray/>



1. Given a collection of intervals, merge all overlapping intervals.

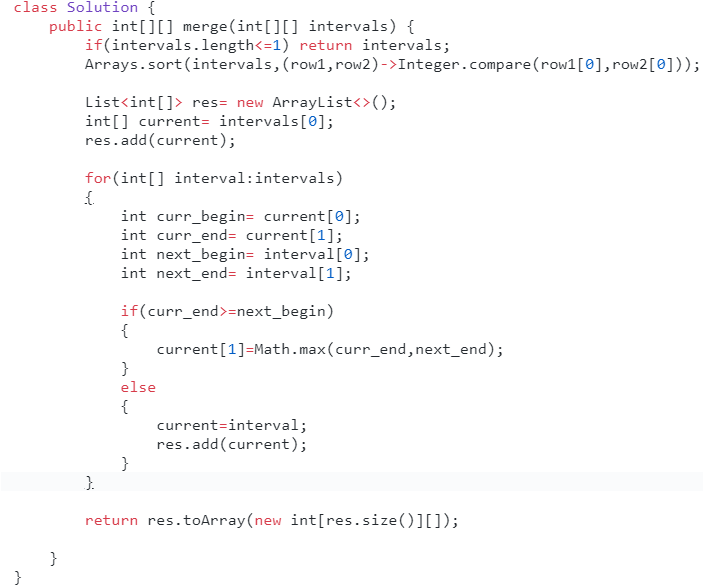
<https://leetcode.com/problems/merge-intervals/solution/>

**Example 1:**

**Input:** [[1,3],[2,6],[8,10],[15,18]]

**Output:** [[1,6],[8,10],[15,18]]

**Explanation:** Since intervals [1,3] and [2,6] overlaps, merge them into [1,6].



1. Given a non-negative integer numRows, generate the first numRows of Pascal's triangle.

**Input:** 5

**Output:**{

[1],

[1,1],

[1,2,1],

[1,3,3,1],

[1,4,6,4,1]

]

