

# Algorithms

## LeetCode OJ

**Mostafa S. Ibrahim**

*Teaching, Training and Coaching for more than a decade!*

*Artificial Intelligence & Computer Vision Researcher*

*PhD from Simon Fraser University - Canada*

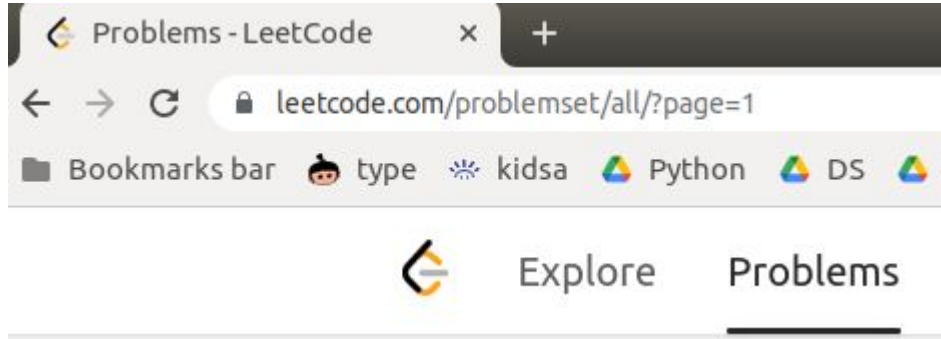
*Bachelor / Msc from Cairo University - Egypt*

*Ex-(Software Engineer / ICPC World Finalist)*



















# Visit: <https://leetcode.com>

- Click The Problems tab



# Problems list

Status	Difficulty	Solution	Acceptance	Difficulty	Frequency
	Easy		31.7%	Hard	
—	Medium		47.4%	Easy	
—	Hard		36.5%	Medium	
—			32.0%	Medium	
—		—	32.3%	Hard	
—			31.1%	Medium	
—			39.3%	Medium	
—			26.1%	Easy	

# 3 Difficulty Levels

- Easy
  - Basic problems to warm up (e.g. search for a number in a sorted array)
  - A good tool to **speed** up coding skills
- Medium
  - These are the core and most important problems
- Hard
  - Out of scope most of the time during the course
    - Also rarely comes in interviews, especially lengthy text or code problems
  - Hard problems are important for competitive programming. But better do hard problems from other OJs (e.g. Codeforces)
- *Label Accuracy: You will encounter some problems with inaccurate labels*

# Filtering Topics

- From tags: you can choose to concentrate exclusively on topics of interest
- In the 'Difficulty' tab, you can sort them from easy to medium to hard
- Notice that Problem #314 is only available for [premium subscription](#)















The screenshot shows the LeetCode website with a 'Filter topics' modal open. The modal contains a search bar 'Filter topics' and a list of topics. The 'Hash Table' and 'Breadth-First Search' topics are highlighted in blue. The 'Expand' link is also visible. The background shows a list of problems with columns for Status, Title, and Difficulty. Problem 314, 'Binary Tree Vertical Order Traversal', is marked as 'Premium' and 'Medium' difficulty.

Status	Title
	132. Palindrome Partitioning II
—	126. Word Ladder II
—	127. Word Ladder
—	133. Clone Graph
	314. Binary Tree Vertical Order Traversal

314. Binary Tree Vertical Order Traversal Plus 48.2% Medium

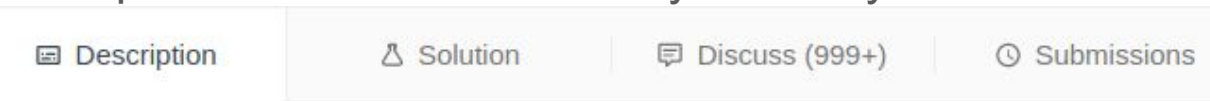
# Solutions & Frequency tabs

- Solutions
  - Some problems have text solutions
  - Some problems have video + text solutions
  - There are also problems without solutions!
  - There are free and 'paid content' solutions
    - There are many solutions on GeeksforGeeks
    - There are also many YouTube videos
- Frequency
  - Subscriber only
  - You can get information on general frequency
  - And even frequency for a specific company

Solution	Frequency
	
	
	
	
	
	
	

# Open a problem

- 4 tabs
- **The Discussions tab** is a great one: you can check if other people have alternative or more succinct solutions. Occasionally, an idea/code in this tab is superior to that of the editorial
  - **Submissions with many languages. Search for your preferred language**
- Avoid problems that are seriously hated by others



## 283. Move Zeroes

Easy



6176



180



Add to List

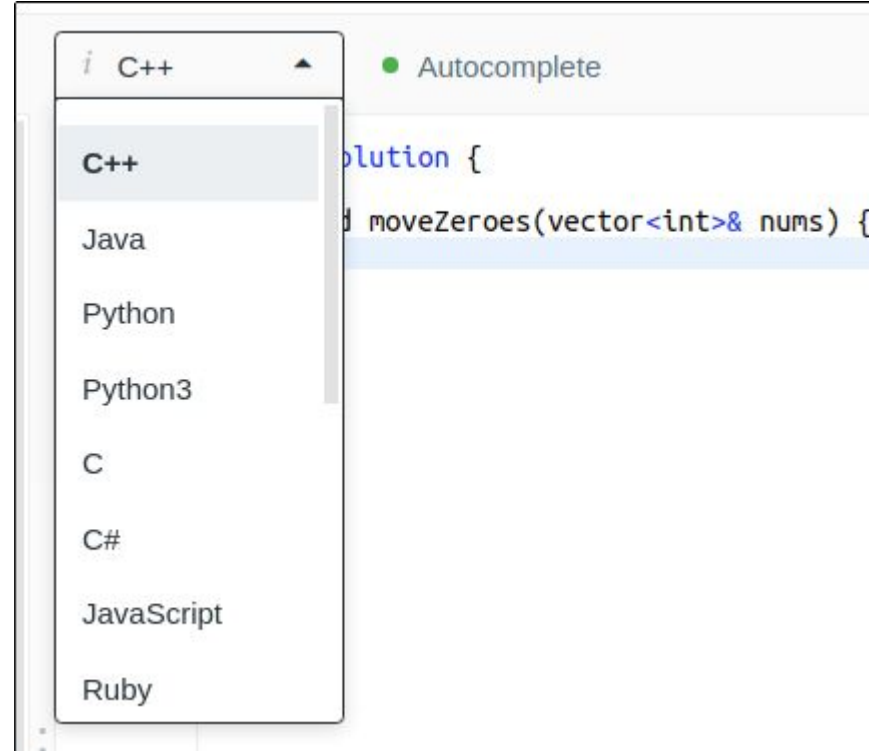


Share

Given an integer array `nums`, move all `0`'s to the end of it while maintaining the relative order of the non-zero elements.

# Problem Solving part

- From the menu, select your language
- It generates the function name
- Fill in this code
- In the submission, never include the main function
- You must use the requested class name





# Problem Solving part: languages

```
class Solution {  
public:  
    void moveZeroes(vector<int>& nums) {  
    }  
};
```

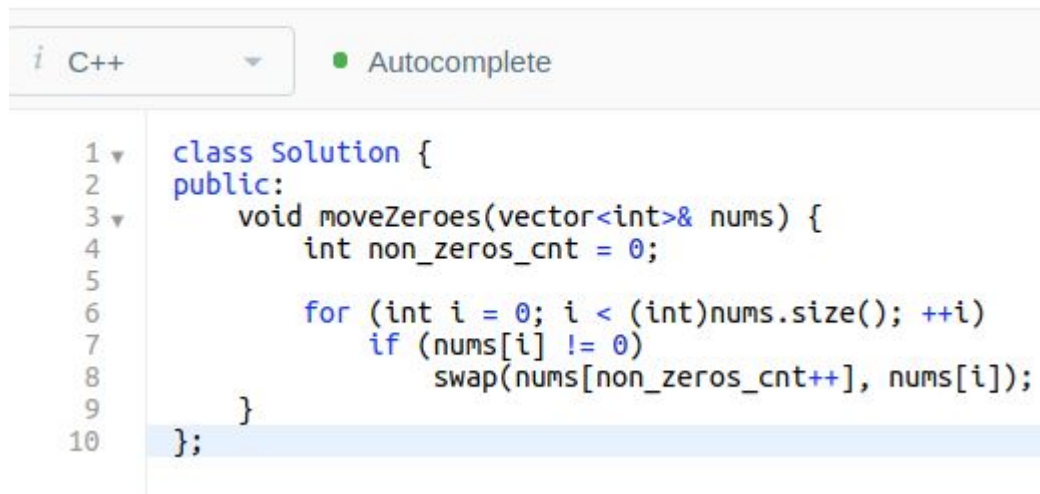
```
public class Solution {  
    public void MoveZeroes(int[] nums) {  
    }  
}
```

```
class Solution:  
    def moveZeroes(self, nums: List[int]) -> None:  
        """  
        Do not return anything, modify nums in-place instead.  
        """
```

```
/**  
 * @param {number[]} nums  
 * @return {void} Do not return anything,  
 * modify nums in-place instead.  
 */  
var moveZeroes = function(nums) {  
};
```

# Problem Solving part: C++ Final code

- No main()
- No printing to the screen
- You don't even need to use #includes
- If they defined a class, don't add to submitted code
- For me
  - I code locally
  - Then rename the main to main1



The screenshot shows a code editor interface. At the top, there is a dropdown menu with 'i C++' and a green 'Autocomplete' button. Below this, the code is displayed with line numbers 1 through 10 on the left. The code defines a class 'Solution' with a public method 'moveZeroes' that takes a vector of integers by reference and moves all zeros to the end of the vector.


```
1 class Solution {  
2 public:  
3     void moveZeroes(vector<int>& nums) {  
4         int non_zeros_cnt = 0;  
5  
6         for (int i = 0; i < (int)nums.size(); ++i)  
7             if (nums[i] != 0)  
8                 swap(nums[non_zeros_cnt++], nums[i]);  
9     }  
10 };
```

# Test and Submit buttons

- Before submitting your code, you can run some of the test cases
  - Copy-paste from the problem.
  - Strictly follow the available style
  - Sometimes it is hard to understand the test case structure

Testcase


Run Code Result

Debugger 

[0,1,0,3,12]

Console ▾

[Use Example Testcases](#)



► Run Code ^

Submit

# Result

- If 'Failed', you can check out the relevant test cases
- If successful, there is a time and memory comparison with other solutions
  - If your code already has good complexity, don't waste much time here

Success [Details >](#)

Runtime: **4 ms**, faster than **89.62%** of C++ online submissions for Move Zeroes.

Memory Usage: **8.7 MB**, less than **98.86%** of C++ online submissions for Move Zeroes.

Next challenges:

[Remove Element](#)

Show off your acceptance:



Time Submitted	Status	Runtime	Memory	Language
08/07/2021 01:33	Accepted	4 ms	8.7 MB	cpp

*“Acquire knowledge and impart it to the people.”*

*“Seek knowledge from the Cradle to the Grave.”*