

Rotate Array - LeetCode

Accepted 10 / 10 testcases passed

LeRKypRx5 submitted at Oct 14, 2025 21:52

Runtime: 8 ms | Beats 28.92% | Memory: 25.82 MB | Beats 8.65%

Runtime graph showing performance across various test cases.

```
class Solution:
    def rotate(self, nums: List[int], k: int) -> None:
        n = len(nums)
        def rotate(start, finish):
            while start <= finish:
                nums[start], nums[finish] = nums[finish], nums[start]
                start += 1
                finish -= 1
            return nums
        rotate(0, len(nums) - 1)
```

Testcase: Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input: nums =

Sort Colors - LeetCode

Accepted 85 / 85 testcases passed

LeRKypRx5 submitted at Oct 15, 2025 09:40

Runtime: 0 ms | Beats 100.00% | Memory: 17.67 MB | Beats 83.23%

Runtime graph showing performance across various test cases.

```
class Solution:
    def sortColors(self, nums: List[int]) -> None:
        n, i = 0, 0
        h = len(nums) - 1
        while n <= h:
            if nums[n] == 0:
                nums[n], nums[h] = nums[h], nums[n]
                h -= 1
            elif nums[n] == 1:
                nums[n], nums[i] = nums[i], nums[n]
                i += 1
```

Testcase: Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input: nums =

Sort Array By Parity - LeetCode

Accepted 285 / 285 testcases passed

LeRKypRx5 submitted at Oct 15, 2025 01:03

Runtime: 0 ms | Beats 100.00% | Memory: 18.38 MB | Beats 82.61%

Runtime graph showing performance across various test cases.

```
def sortArrayByParity(self, nums: List[int]) -> List[int]:
    j = 0
    for i in range(0, len(nums)):
        if nums[i] % 2 == 0:
            nums[i], nums[j] = nums[j], nums[i]
            j += 1
    return nums
```

Testcase: Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input: nums =

Move Zeroes - LeetCode

Accepted 75 / 75 testcases passed
LefRkypRxS submitted at Oct 15, 2025 01:08

Runtime: 0 ms | Beats 100.00%
Memory: 19.04 MB | Beats 9.76%

Code

```
class Solution:
    def moveZeroes(self, nums: List[int]) -> None:
        index = 0
        for i in range(len(nums)):
            if nums[i] != 0:
                nums[i], nums[index] = nums[index], nums[i]
                index += 1
        return nums
```

Testcase

Input: [0, 1, 0, 3, 12]

Output: [1, 3, 12, 0, 0]

Merge Sorted Array - LeetCode

Accepted 59 / 59 testcases passed
LefRkypRxS submitted at Oct 15, 2025 01:10

Runtime: 0 ms | Beats 100.00%
Memory: 17.72 MB | Beats 71.86%

Code

```
class Solution:
    def merge(self, nums1: List[int], m: int, nums2: List[int], n: int) -> None:
        p1 = m - 1
        p2 = n - 1
        p3 = m + n - 1
        while p2 >= 0 and p3 >= 0:
            if nums1[p2] >= nums2[p3]:
                nums1[p3] = nums1[p2]
                p2 -= 1
            else:
                nums1[p3] = nums2[p3]
                p3 -= 1
        p3 -= 1
```

Testcase

Input: [1, 2, 3, 4, 5], [6, 7, 8, 9, 10]

Output: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Two Sum II - Input Array Is Sorted - LeetCode

Accepted 14 / 14 testcases passed
LefRkypRxS submitted at Oct 15, 2025 01:11

Runtime: 3 ms | Beats 81.09%
Memory: 18.54 MB | Beats 59.38%

Code

```
class Solution:
    def twoSum(self, numbers: List[int], target: int) -> List[int]:
        n = len(numbers)
        l, r = 0, n - 1
        while l < r:
            current_sum = numbers[l] + numbers[r]
            if current_sum == target:
                return [l + 1, r + 1]
            elif current_sum < target:
                l += 1
            else:
                r -= 1
```

Testcase

Case 1

Input: [2, 7, 11, 15], target = 9

Output: [1, 2]