

Aditya Pratap Singh

B.Tech CS (Hons.) -EA

2315800005

**Question1.** [**Sort an Array**](https://leetcode.com/problems/sort-an-array/) **(Leetcode-Medium-912)**

Given an array of integers nums, sort the array in ascending order and return it.

You must solve the problem without using any built-in functions in O(nlog(n)) time complexity and with the smallest space complexity possible.

**Example 1:**

**Input:** nums = [5,2,3,1]

**Output:** [1,2,3,5]

**Explanation:** After sorting the array, the positions of some numbers are not changed (for example, 2 and 3), while the positions of other numbers are changed (for example, 1 and 5).

**Example 2:**

**Input:** nums = [5,1,1,2,0,0]

**Output:** [0,0,1,1,2,5]

**Explanation:** Note that the values of nums are not necessarily unique.

**Constraints:**

* 1 <= nums.length <= 5 \* 104
* -5 \* 104 <= nums[i] <= 5 \* 104

**Topics**

[Array](https://leetcode.com/tag/array/)[Divide and Conquer](https://leetcode.com/tag/divide-and-conquer/)[Sorting](https://leetcode.com/tag/sorting/)[Heap (Priority Queue)](https://leetcode.com/tag/heap-priority-queue/)[Merge Sort](https://leetcode.com/tag/merge-sort/)[Bucket Sort](https://leetcode.com/tag/bucket-sort/)[Radix Sort](https://leetcode.com/tag/radix-sort/)[Counting Sort](https://leetcode.com/tag/counting-sort/)

[**Question2. Merge Sorted Array**](https://leetcode.com/problems/merge-sorted-array/) **(Leetcode-Medium-88)**

You are given two integer arrays nums1 and nums2, sorted in non-decreasing order, and two integers m and n, representing the number of elements in nums1 and nums2 respectively.

Merge nums1 and nums2 into a single array sorted in non-decreasing order.

The final sorted array should not be returned by the function, but instead be *stored inside the array*nums1. To accommodate this, nums1 has a length of m + n, where the first m elements denote the elements that should be merged, and the last n elements are set to 0 and should be ignored. nums2 has a length of n.

**Example 1:**

**Input:** nums1 = [1,2,3,0,0,0], m = 3, nums2 = [2,5,6], n = 3

**Output:** [1,2,2,3,5,6]

**Explanation:** The arrays we are merging are [1,2,3] and [2,5,6].

The result of the merge is [1,2,2,3,5,6] with the underlined elements coming from nums1.

**Example 2:**

**Input:** nums1 = [1], m = 1, nums2 = [], n = 0

**Output:** [1]

**Explanation:** The arrays we are merging are [1] and [].

The result of the merge is [1].

**Example 3:**

**Input:** nums1 = [0], m = 0, nums2 = [1], n = 1

**Output:** [1]

**Explanation:** The arrays we are merging are [] and [1].

The result of the merge is [1].

Note that because m = 0, there are no elements in nums1. The 0 is only there to ensure the merge result can fit in nums1.

**Constraints:**

* nums1.length == m + n
* nums2.length == n
* 0 <= m, n <= 200
* 1 <= m + n <= 200
* -109 <= nums1[i], nums2[j] <= 109

**Follow up:**Can you come up with an algorithm that runs in O(m + n) time?

Topics

[Array](https://leetcode.com/tag/array/)[Two Pointers](https://leetcode.com/tag/two-pointers/)[Sorting](https://leetcode.com/tag/sorting/)

[**Question3. Majority Element**](https://leetcode.com/problems/majority-element/) **(Leetcode-Easy-169)**

Given an array nums of size n, return *the majority element*.

The majority element is the element that appears more than ⌊n / 2⌋ times. You may assume that the majority element always exists in the array.

**Example 1:**

**Input:** nums = [3,2,3]

**Output:** 3

**Example 2:**

**Input:** nums = [2,2,1,1,1,2,2]

**Output:** 2

**Constraints:**

* n == nums.length
* 1 <= n <= 5 \* 104
* -109 <= nums[i] <= 109

**Follow-up:** Could you solve the problem in linear time and in O(1) space?

Topics

[Array](https://leetcode.com/tag/array/)[Hash Table](https://leetcode.com/tag/hash-table/)[Divide and Conquer](https://leetcode.com/tag/divide-and-conquer/)[Sorting](https://leetcode.com/tag/sorting/)[Counting](https://leetcode.com/tag/counting/)

**Question4. Array Search (GFG-Basic)**

Given an array, **arr[]** of n integers, and an integer element **x**, find whether element **x** is present in the array. Return the index of the **first occurrence** of **x** in the array, or -1 if it doesn't exist.

**Examples:**

**Input:** arr[] = [1, 2, 3, 4], x = 3  
**Output:** 2

**Explanation:** For array [1, 2, 3, 4], the element to be searched is 3. Since 3 is present at index 2, the output is 2.

**Input:** arr[] = [10, 8, 30, 4, 5], x = 5  
**Output:** 4

**Explanation:** For array [10, 8, 30, 4, 5], the element to be searched is 5 and it is at index 4. So, the output is 4.

**Input:** arr[] = [10, 8, 30], x = 6  
**Output:** -1

**Explanation:** The element to be searched is 6 and it is not present, so we return -1.

**Constraints:**  
1 ≤ arr.size ≤ 106  
0 ≤ arr[i] ≤ 106  
0 ≤ x ≤ 105**Topic Tags**

[Arrays](https://www.geeksforgeeks.org/explore?category%5b%5d=Arrays)[Searching](https://www.geeksforgeeks.org/explore?category%5b%5d=Searching)[Algorithms](https://www.geeksforgeeks.org/explore?category%5b%5d=Algorithms)

**Question5. Binary Search** **(Leetcode-Medium-704)**

Given an array of integers nums which is sorted in ascending order, and an integer target, write a function to search target in nums. If target exists, then return its index. Otherwise, return -1.

You must write an algorithm with O(log n) runtime complexity.

**Example 1:**

**Input:** nums = [-1,0,3,5,9,12], target = 9

**Output:** 4

**Explanation:** 9 exists in nums and its index is 4

**Example 2:**

**Input:** nums = [-1,0,3,5,9,12], target = 2

**Output:** -1

**Explanation:** 2 does not exist in nums so return -1

**Constraints:**

* 1 <= nums.length <= 104
* -104 < nums[i], target < 104
* All the integers in nums are **unique**.
* nums is sorted in ascending order.

Topics

[Array](https://leetcode.com/tag/array/)[Binary Search](https://leetcode.com/tag/binary-search/)

[**Question6. Search Insert Position**](https://leetcode.com/problems/search-insert-position/)**(Leetcode-Medium-35)**

Given a sorted array of distinct integers and a target value, return the index if the target is found. If not, return the index where it would be if it were inserted in order.

You must write an algorithm with O(log n) runtime complexity.

**Example 1:**

**Input:** nums = [1,3,5,6], target = 5

**Output:** 2

**Example 2:**

**Input:** nums = [1,3,5,6], target = 2

**Output:** 1

**Example 3:**

**Input:** nums = [1,3,5,6], target = 7

**Output:** 4

**Constraints:**

* 1 <= nums.length <= 104
* -104 <= nums[i] <= 104
* nums contains **distinct** values sorted in **ascending** order.
* -104 <= target <= 104

Topics

[Array](https://leetcode.com/tag/array/)[Binary Search](https://leetcode.com/tag/binary-search/)

[**Question7. Find First and Last Position of Element in Sorted Array**](https://leetcode.com/problems/find-first-and-last-position-of-element-in-sorted-array/)**(Leetcode-Medium-35)**

Given an array of integers nums sorted in non-decreasing order, find the starting and ending position of a given target value.If target is not found in the array, return [-1, -1].You must write an algorithm with O(log n) runtime complexity.

**Example 1:**

**Input:** nums = [5,7,7,8,8,10], target = 8

**Output:** [3,4]

**Example 2:**

**Input:** nums = [5,7,7,8,8,10], target = 6

**Output:** [-1,-1]

**Example 3:**

**Input:** nums = [], target = 0

**Output:** [-1,-1]

**Constraints:**

* 0 <= nums.length <= 105
* -109 <= nums[i] <= 109
* nums is a non-decreasing array.
* -109 <= target <= 109

Topics

[Array](https://leetcode.com/tag/array/)[Binary Search](https://leetcode.com/tag/binary-search/)