

Problem 912: Sort an Array

Problem Information

Difficulty: Medium

Acceptance Rate: 56.06%

Paid Only: No

Tags: Array, Divide and Conquer, Sorting, Heap (Priority Queue), Merge Sort, Bucket Sort, Radix Sort, Counting Sort

Problem Description

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 $O(n \log(n))$ 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:

$1 \leq \text{nums.length} \leq 5 \cdot 10^4$ $-5 \cdot 10^4 \leq \text{nums}[i] \leq 5 \cdot 10^4$

Code Snippets

C++:

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

Java:

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

Python3:

```
class Solution:  
    def sortArray(self, nums: List[int]) -> List[int]:
```