Problem 315: Count of Smaller Numbers After Self

Problem Information

Difficulty: Hard
Acceptance Rate: 43.14%
Paid Only: No
Tags: Array, Binary Search, Divide and Conquer, Binary Indexed Tree, Segment Tree, Merge Sort, Ordered Set

Problem Description

Given an integer array `nums`, return _an integer array_`counts` _where_`counts[i]`_is the number of smaller elements to the right of_`nums[i]`.

Example 1:

Input: nums = [5,2,6,1] **Output:** [2,1,1,0] **Explanation:** To the right of 5 there are **2** smaller elements (2 and 1). To the right of 2 there is only **1** smaller element (1). To the right of 6 there is **1** smaller element (1). To the right of 1 there is **0** smaller element.

Example 2:

Input: nums = [-1] **Output:** [0]

Example 3:

Input: nums = [-1,-1] **Output:** [0,0]

Constraints:

* `1 <= nums.length <= 105` * `-104 <= nums[i] <= 104`

Code Snippets

C++:

```
class Solution {
public:
vector<int> countSmaller(vector<int>& nums) {

}
};
```

Java:

```
class Solution {
public List<Integer> countSmaller(int[] nums) {

}
}
```

Python3:

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