

Problem 891: Sum of Subsequence Widths

Problem Information

Difficulty: Hard

Acceptance Rate: 39.80%

Paid Only: No

Tags: Array, Math, Sorting

Problem Description

The **width** of a sequence is the difference between the maximum and minimum elements in the sequence.

Given an array of integers `nums`, return the sum of the **widths** of all the non-empty **subsequences** of `nums`. Since the answer may be very large, return it **modulo** $10^9 + 7$.

A **subsequence** is a sequence that can be derived from an array by deleting some or no elements without changing the order of the remaining elements. For example, `[3,6,2,7]` is a subsequence of the array `[0,3,1,6,2,2,7]`.

Example 1:

Input: `nums = [2,1,3]` **Output:** 6 **Explanation:** The subsequences are `[1]`, `[2]`, `[3]`, `[2,1]`, `[2,3]`, `[1,3]`, `[2,1,3]`. The corresponding widths are 0, 0, 0, 1, 1, 2, 2. The sum of these widths is 6.

Example 2:

Input: `nums = [2]` **Output:** 0

Constraints:

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

Code Snippets

C++:

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

Java:

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

Python3:

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