5212. Sum of Floored Pairs

My Submissions (/contest/biweekly-contest-52/problems/sum-of-floored-pairs/submissions/)

Back to Contest (/contest/biweekly-contest-52/)

Given an integer array `nums` , return the sum of `floor(nums[i] / nums[j])` for all pairs of indices `0 <= i, j < nums.length` in the array. Since the answer may be too large, return it **modulo** $10^9 + 7$.

The `floor()` function returns the integer part of the division.

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Hard

Example 1:

```
Input: nums = [2,5,9]
Output: 10
Explanation:
floor(2 / 5) = floor(2 / 9) = floor(5 / 9) = 0
floor(2 / 2) = floor(5 / 5) = floor(9 / 9) = 1
floor(5 / 2) = 2
floor(9 / 2) = 4
floor(9 / 5) = 1
We calculate the floor of the division for every pair of indices in the array th
```

Example 2:

```
Input: nums = [7,7,7,7,7,7,7]
Output: 49
```

Constraints:

- `1 <= nums.length <= 10^5`
- `1 <= nums[i] <= 10^5`

JavaScript

```
const N = 3e5;
const MOD = 1e9 + 7;
const sumOfFlooredPairs = (a) => {
    let n = a.length;
    let freq  = Array(N).fill(0);
    let preFreq  = Array(N).fill(0);
    for (let  i = 0; i < n; i++) freq[a[i]]++;
    for (let i = 1; i < N; i++) preFreq[i] = preFreq[i - 1] + freq[i];
    let  res = 0;
    for (let  i = 1; i < N; i++) {
        for (let  j = i; j < N; j += i) {
            let X = (preFreq[j - 1] - preFreq[Math.abs(j - i - 1)]);
            res += X * ((j / i >> 0) - 1) * freq[i];
        }
    }
    return res % MOD;
};
```

Custom Testcase Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/493505314/) More Details (/submissions/detail/493505314/)

Share your acceptance!