Problem 2615: Sum of Distances

Problem Information

Difficulty: Medium
Acceptance Rate: 32.07%
Paid Only: No
Tags: Array, Hash Table, Prefix Sum

Problem Description

You are given a **0-indexed** integer array `nums`. There exists an array `arr` of length `nums.length`, where `arr[i]` is the sum of `|i - j|` over all `j` such that `nums[j] == nums[i]` and `j != i`. If there is no such `j`, set `arr[i]` to be `0`.

Return _the array_`arr` _._

Example 1:

Input: nums = [1,3,1,1,2] **Output:** [5,0,3,4,0] **Explanation:** When i = 0, nums[0] == nums[2] and nums[0] == nums[3]. Therefore, arr[0] = |0 - 2| + |0 - 3| = 5. When i = 1, arr[1] = 0 because there is no other index with value 3. When i = 2, nums[2] == nums[0] and nums[2] == nums[3]. Therefore, arr[2] = |2 - 0| + |2 - 3| = 3. When i = 3, nums[3] == nums[0] and nums[3] == nums[2]. Therefore, arr[3] = |3 - 0| + |3 - 2| = 4. When i = 4, arr[4] = 0 because there is no other index with value 2.

Example 2:

Input: nums = [0,5,3] **Output:** [0,0,0] **Explanation:** Since each element in nums is distinct, arr[i] = 0 for all i.

Constraints:

* `1 <= nums.length <= 105` * `0 <= nums[i] <= 109`

Note: This question is the same as [2121: Intervals Between Identical Elements.](https://leetcode.com/problems/intervals-between-identical- elements/description/)

Code Snippets

C++:

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

}
};
```

Java:

```
class Solution {
public long[] distance(int[] nums) {

}
}
```

Python3:

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