Problem 3139: Minimum Cost to Equalize Array

Problem Information

Difficulty: Hard
Acceptance Rate: 18.20%
Paid Only: No
Tags: Array, Greedy, Enumeration

Problem Description

You are given an integer array `nums` and two integers `cost1` and `cost2`. You are allowed to perform **either** of the following operations **any** number of times:

* Choose an index `i` from `nums` and **increase** `nums[i]` by `1` for a cost of `cost1`. * Choose two **different** indices `i`, `j`, from `nums` and **increase** `nums[i]` and `nums[j]` by `1` for a cost of `cost2`.

Return the **minimum** **cost** required to make all elements in the array **equal** _._

Since the answer may be very large, return it **modulo** `109 + 7`.

Example 1:

Input: nums = [4,1], cost1 = 5, cost2 = 2

Output: 15

Explanation:

The following operations can be performed to make the values equal:

* Increase `nums[1]` by 1 for a cost of 5. `nums` becomes `[4,2]`. * Increase `nums[1]` by 1 for a cost of 5. `nums` becomes `[4,3]`. * Increase `nums[1]` by 1 for a cost of 5. `nums` becomes `[4,4]`.

The total cost is 15.

Example 2:

Input: nums = [2,3,3,3,5], cost1 = 2, cost2 = 1

Output: 6

Explanation:

The following operations can be performed to make the values equal:

* Increase `nums[0]` and `nums[1]` by 1 for a cost of 1. `nums` becomes `[3,4,3,3,5]`. * Increase `nums[0]` and `nums[2]` by 1 for a cost of 1. `nums` becomes `[4,4,4,3,5]`. * Increase `nums[0]` and `nums[3]` by 1 for a cost of 1. `nums` becomes `[5,4,4,4,5]`. * Increase `nums[1]` and `nums[2]` by 1 for a cost of 1. `nums` becomes `[5,5,5,4,5]`. * Increase `nums[3]` by 1 for a cost of 2. `nums` becomes `[5,5,5,5,5]`.

The total cost is 6.

Example 3:

Input: nums = [3,5,3], cost1 = 1, cost2 = 3

Output: 4

Explanation:

The following operations can be performed to make the values equal:

* Increase `nums[0]` by 1 for a cost of 1. `nums` becomes `[4,5,3]`. * Increase `nums[0]` by 1 for a cost of 1. `nums` becomes `[5,5,3]`. * Increase `nums[2]` by 1 for a cost of 1. `nums` becomes `[5,5,4]`. * Increase `nums[2]` by 1 for a cost of 1. `nums` becomes `[5,5,5]`.

The total cost is 4.

Constraints:

* `1 <= nums.length <= 105` * `1 <= nums[i] <= 106` * `1 <= cost1 <= 106` * `1 <= cost2 <= 106`

Code Snippets

C++:

```
class Solution {
public:
int minCostToEqualizeArray(vector<int>& nums, int cost1, int cost2) {

}
};
```

Java:

```
class Solution {
public int minCostToEqualizeArray(int[] nums, int cost1, int cost2) {

}
}
```

Python3:

```
class Solution:
def minCostToEqualizeArray(self, nums: List[int], cost1: int, cost2: int) ->
int:
```