3139. Minimum Cost to Equalize Array

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** `10^9 + 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 <= 10^5`
- `1 <= nums[i] <= 10^6`
- `1 <= cost1 <= 10^6`
- `1 <= cost2 <= 10^6`