

# 2448. Minimum Cost to Make Array Equal

## Description

You are given two **0-indexed** arrays `nums` and `cost` consisting each of `n` **positive** integers.

You can do the following operation **any** number of times:

- Increase or decrease **any** element of the array `nums` by `1`.

The cost of doing one operation on the `ith` element is `cost[i]`.

Return *the **minimum** total cost such that all the elements of the array `nums` become **equal***.

### Example 1:

**Input:** `nums = [1,3,5,2]`, `cost = [2,3,1,14]`

**Output:** 8

**Explanation:** We can make all the elements equal to 2 in the following way:

- Increase the `0th` element one time. The cost is 2.
- Decrease the `1st` element one time. The cost is 3.
- Decrease the `2nd` element three times. The cost is  $1 + 1 + 1 = 3$ .

The total cost is  $2 + 3 + 3 = 8$ .

It can be shown that we cannot make the array equal with a smaller cost.

### Example 2:

**Input:** `nums = [2,2,2,2,2]`, `cost = [4,2,8,1,3]`

**Output:** 0

**Explanation:** All the elements are already equal, so no operations are needed.

### Constraints:

- `n == nums.length == cost.length`
- `1 <= n <= 105`
- `1 <= nums[i], cost[i] <= 106`
- Test cases are generated in a way that the output doesn't exceed  $2^{53} - 1$

