

2567. Minimum Score by Changing Two Elements

Description

You are given a **0-indexed** integer array `nums`.

- The **low** score of `nums` is the minimum value of `|nums[i] - nums[j]|` over all `0 <= i < j < nums.length`.
- The **high** score of `nums` is the maximum value of `|nums[i] - nums[j]|` over all `0 <= i < j < nums.length`.
- The **score** of `nums` is the sum of the **high** and **low** scores of `nums`.

To minimize the score of `nums`, we can change the value of **at most two** elements of `nums`.

Return *the minimum possible score after changing the value of at most two elements of* `nums`.

Note that `|x|` denotes the absolute value of `x`.

Example 1:

Input: `nums = [1,4,3]`

Output: 0

Explanation: Change value of `nums[1]` and `nums[2]` to 1 so that `nums` becomes `[1,1,1]`. Now, the value of `|nums[i] - nums[j]|` is always equal to 0, so we return `0 + 0 = 0`.

Example 2:

Input: `nums = [1,4,7,8,5]`

Output: 3

Explanation: Change `nums[0]` and `nums[1]` to be 6. Now `nums` becomes `[6,6,7,8,5]`.

Our low score is achieved when `i = 0` and `j = 1`, in which case `|nums[i] - nums[j]| = |6 - 6| = 0`.

Our high score is achieved when `i = 3` and `j = 4`, in which case `|nums[i] - nums[j]| = |8 - 5| = 3`.

The sum of our high and low score is 3, which we can prove to be minimal.

Constraints:

- `3 <= nums.length <= 105`
- `1 <= nums[i] <= 109`

