Problem 908: Smallest Range I

Problem Information

Difficulty: Easy
Acceptance Rate: 72.88%
Paid Only: No
Tags: Array, Math

Problem Description

You are given an integer array `nums` and an integer `k`.

In one operation, you can choose any index `i` where `0 <= i < nums.length` and change `nums[i]` to `nums[i] + x` where `x` is an integer from the range `[-k, k]`. You can apply this operation **at most once** for each index `i`.

The **score** of `nums` is the difference between the maximum and minimum elements in `nums`.

Return _the minimum**score** of _`nums` _after applying the mentioned operation at most once for each index in it_.

Example 1:

Input: nums = [1], k = 0 **Output:** 0 **Explanation:** The score is max(nums) - min(nums) = 1 - 1 = 0.

Example 2:

Input: nums = [0,10], k = 2 **Output:** 6 **Explanation:** Change nums to be [2, 8]. The score is max(nums) - min(nums) = 8 - 2 = 6.

Example 3:

Input: nums = [1,3,6], k = 3 **Output:** 0 **Explanation:** Change nums to be [4, 4, 4]. The score is max(nums) - min(nums) = 4 - 4 = 0.

Constraints:

* `1 <= nums.length <= 104` * `0 <= nums[i] <= 104` * `0 <= k <= 104`

Code Snippets

C++:

```
class Solution {
public:
int smallestRangeI(vector<int>& nums, int k) {

}
};
```

Java:

```
class Solution {
public int smallestRangeI(int[] nums, int k) {

}
}
```

Python3:

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