# 题目

给你一个整数数组 nums，和一个整数 k 。

在一个操作中，您可以选择0 <= i < nums.length的任何索引i。将nums[i]改为nums[i] + x，其中x是一个范围为[-k, k]的整数。对于每个索引i，最多只能应用一次此操作。

nums的分数是nums中最大和最小元素的差值。

在对nums中的每个索引最多应用一次上述操作后，返回 nums 的最低 分数 。

示例 1：

输入：nums = [1], k = 0

输出：0

解释：分数是 max(nums) - min(nums) = 1 - 1 = 0。

示例 2：

输入：nums = [0,10], k = 2

输出：6

解释：将 nums 改为 [2,8]。分数是 max(nums) - min(nums) = 8 - 2 = 6。

示例 3：

输入：nums = [1,3,6], k = 3

输出：0

解释：将 nums 改为 [4,4,4]。分数是 max(nums) - min(nums) = 4 - 4 = 0。

提示：

1 <= nums.length <= 10^4

0 <= nums[i] <= 10^4

0 <= k <= 10^4

# 分析

## 方法一：模拟

思路：

首先找到数组中的最小值和最大值，然后计算它们之间的差值减去2 \* k的结果，如果结果小于 0，则返回 0，否则返回该结果。

代码：

class Solution {

public:

int smallestRangeI(vector<int>& nums, int k) {

int minNum = INT\_MAX, maxNum = INT\_MIN;

for (int num : nums) {

minNum = min(minNum, num);

maxNum = max(maxNum, num);

}

int diff = maxNum - minNum - 2 \* k;

// maxNum – k – (minNum + k)，即最大值减去k，最小值加上k，“逼近”思想

return max(0, diff);

}

};