# 题目

给你一个下标从 0 开始的整数数组 nums 和一个整数 k 。

一次操作中，你可以删除 nums 中的最小元素。

你需要使数组中的所有元素都大于或等于 k ，请你返回需要的最少操作次数。

示例 1：

输入：nums = [2,11,10,1,3], k = 10

输出：3

解释：第一次操作后，nums 变为 [2, 11, 10, 3] 。

第二次操作后，nums 变为 [11, 10, 3] 。

第三次操作后，nums 变为 [11, 10] 。

此时，数组中的所有元素都大于等于 10 ，所以我们停止操作。

使数组中所有元素都大于等于 10 需要的最少操作次数为 3 。

示例 2：

输入：nums = [1,1,2,4,9], k = 1

输出：0

解释：数组中的所有元素都大于等于 1 ，所以不需要对 nums 做任何操作。

示例 3：

输入：nums = [1,1,2,4,9], k = 9

输出：4

解释：nums 中只有一个元素大于等于 9 ，所以需要执行 4 次操作。

提示：

1 <= nums.length <= 50

1 <= nums[i] <= 10^9

1 <= k <= 10^9

输入保证至少有一个满足 nums[i] >= k 的下标 i 存在。

# 分析

## 方法一：遍历

思路：

由于所有小于k的数都要被删除，所以统计nums中的小于k的元素个数即可。

代码：

class Solution {

public:

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

int ans = 0;

for (int x : nums) {

ans += x < k;

}

return ans;

}

};

复杂度分析

时间复杂度：O(n)，其中n为 nums的长度。

空间复杂度：O(1)。