# 题目

给你一个 非负 整数数组nums和一个整数k。

如果一个数组中所有元素的按位或运算OR的值至少为k，那么我们称这个数组是特别的。

请你返回nums中最短特别非空子数组的长度，如果特别子数组不存在，那么返回-1 。

示例 1：

输入：nums = [1,2,3], k = 2

输出：1

解释：

子数组 [3] 的按位 OR 值为 3 ，所以我们返回 1 。

示例 2：

输入：nums = [2,1,8], k = 10

输出：3

解释：

子数组 [2,1,8] 的按位 OR 值为 11 ，所以我们返回 3 。

示例 3：

输入：nums = [1,2], k = 0

输出：1

解释：

子数组 [1] 的按位 OR 值为 1 ，所以我们返回 1。

提示：

1 <= nums.length <= 50

0 <= nums[i] <= 50

0 <= k < 64

# 分析

你可以使用滑动窗口来解决这个问题。维护一个窗口，使得窗口内的子数组的按位或运算结果至少为 k，并且窗口的长度最小。具体做法如下：

class Solution {

public:

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

int n = nums.size();

int left = 0, right = 0;

int result = INT\_MAX;

int currentOR = 0;

while (right < n) {

currentOR |= nums[right];

while (currentOR >= k) {

result = min(result, right - left + 1);

currentOR &= ~nums[left];

left++;

}

right++;

}

return result == INT\_MAX ? -1 : result;

}

};

这段代码使用了一个滑动窗口来维护符合条件的子数组。首先，right 指针向右移动，不断扩大窗口，直到窗口内的元素按位或的结果不小于 k。然后，left 指针向右移动，缩小窗口，直到窗口内的元素按位或的结果小于 k。在这个过程中不断更新结果。