# 题目

给你一个长度为n的整数数组nums 。

考虑nums中进行按位与（bitwise AND）运算得到的值最大的非空子数组。

换句话说，令k是nums任意子数组执行按位与运算所能得到的最大值。那么，只需要考虑那些执行一次按位与运算后等于 k 的子数组。

返回满足要求的 最长 子数组的长度。

数组的按位与就是对数组中的所有数字进行按位与运算。

子数组 是数组中的一个连续元素序列。

示例 1：

输入：nums = [1,2,3,3,2,2]

输出：2

解释：

子数组按位与运算的最大值是 3 。

能得到此结果的最长子数组是 [3,3]，所以返回 2 。

示例 2：

输入：nums = [1,2,3,4]

输出：1

解释：

子数组按位与运算的最大值是 4 。

能得到此结果的最长子数组是 [4]，所以返回 1 。

提示：

1 <= nums.length <= 105

1 <= nums[i] <= 106

# 分析

class Solution {

public:

int longestSubarray(vector<int>& nums) {

int ans = 0, max = 0, cnt = 0;

for (int i = 0; i < nums.size(); i++) {

if (nums[i] > max) {

max = nums[i];

ans = cnt = 1;

} else if (nums[i] == max) {

cnt++;

if (cnt > ans) ans = cnt;

} else {

cnt = 0;

}

}

return ans;

}

};