# 题目

给你一个整数数组 nums ，和一个表示限制的整数 limit，请你返回最长连续子数组的长度，该子数组中的任意两个元素之间的绝对差必须小于或者等于 limit 。

如果不存在满足条件的子数组，则返回 0 。

示例 1：

输入：nums = [8,2,4,7], limit = 4

输出：2

解释：所有子数组如下：

[8] 最大绝对差 |8-8| = 0 <= 4.

[8,2] 最大绝对差 |8-2| = 6 > 4.

[8,2,4] 最大绝对差 |8-2| = 6 > 4.

[8,2,4,7] 最大绝对差 |8-2| = 6 > 4.

[2] 最大绝对差 |2-2| = 0 <= 4.

[2,4] 最大绝对差 |2-4| = 2 <= 4.

[2,4,7] 最大绝对差 |2-7| = 5 > 4.

[4] 最大绝对差 |4-4| = 0 <= 4.

[4,7] 最大绝对差 |4-7| = 3 <= 4.

[7] 最大绝对差 |7-7| = 0 <= 4.

因此，满足题意的最长子数组的长度为 2 。

示例 2：

输入：nums = [10,1,2,4,7,2], limit = 5

输出：4

解释：满足题意的最长子数组是 [2,4,7,2]，其最大绝对差 |2-7| = 5 <= 5 。

示例 3：

输入：nums = [4,2,2,2,4,4,2,2], limit = 0

输出：3

提示：

1 <= nums.length <= 10^5

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

0 <= limit <= 10^9

说明：

与2762. 不间断子数组类似

# 分析

分析：

代码：

class Solution {

public:

int longestSubarray(vector<int>& nums, int limit) {

int ans = 0;

multiset<int> s;

int left = 0, right = 0, n = nums.size();

for (; right < n; right++) {

s.insert(nums[right]);

if (abs(\*s.rbegin() - \*s.begin()) > limit) {

auto pos = s.find(nums[left++]);

s.erase(pos);

}

int len = right - left + 1;

if (len > ans) ans = len;

}

return ans;

}

};