# 题目

给你一个长度为 n 的整数数组 nums ，请你返回 nums 中最 接近 0 的数字。如果有多个答案，请你返回它们中的 最大值 。

示例 1：

输入：nums = [-4,-2,1,4,8]

输出：1

解释：

-4 到 0 的距离为 |-4| = 4 。

-2 到 0 的距离为 |-2| = 2 。

1 到 0 的距离为 |1| = 1 。

4 到 0 的距离为 |4| = 4 。

8 到 0 的距离为 |8| = 8 。

所以，数组中距离 0 最近的数字为 1 。

示例 2：

输入：nums = [2,-1,1]

输出：1

解释：1 和 -1 都是距离 0 最近的数字，所以返回较大值 1 。

提示：

1 <= n <= 1000

-10^5 <= nums[i] <= 10^5

# 分析

## 方法一：直接遍历

思路：

代码：

class Solution {

public:

int findClosestNumber(vector<int>& nums) {

int result = nums[0];

for (auto& num:nums)

/\* nums中最接近0的数字，如果距离相等记录较大的 \*/

if(abs(num) < abs(result) || (abs(num) == abs(result) && num > result))

result = num;

return result;

}

};

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)。