# 题目

给你一个整数数组nums，其中总是存在唯一的一个最大整数。

请你找出数组中的最大元素并检查它是否至少是数组中每个其他数字的两倍。如果是，则返回最大元素的下标，否则返回-1。

示例 1：

输入：nums = [3,6,1,0]

输出：1

解释：6是最大的整数，对于数组中的其他整数，6至少是数组中其他元素的两倍。6的下标是1，所以返回1。

示例 2：

输入：nums = [1,2,3,4]

输出：-1

解释：4没有超过3的两倍大，所以返回-1。

提示：

2 <= nums.length <= 50

0 <= nums[i] <= 100

nums 中的最大元素是唯一的

# 分析

## 方法一：遍历

思路：

遍历数组分别找到数组的最大值m1和次大值m2。如果m1≥m2×2成立，则最大值至少是数组其余数字的两倍，此时返回最大值的下标，否则返回-1。

为了返回最大值的下标，我们需要在计算最大值的同时记录最大值的下标。

代码：

class Solution {

public:

int dominantIndex(vector<int>& nums) {

int m1 = -1, m2 = -1;

int index = -1;

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

if (nums[i] > m1) {

m2 = m1;

m1 = nums[i];

index = i;

} else if (nums[i] > m2) {

m2 = nums[i];

}

}

return m1 >= m2 \* 2 ? index : -1;

}

};