# 题目

给你一个整数数组 nums ，数组由 不同正整数 组成，请你找出并返回数组中 任一 既不是 最小值 也不是 最大值 的数字，如果不存在这样的数字，返回 -1 。

返回所选整数。

示例 1：

输入：nums = [3,2,1,4]

输出：2

解释：在这个示例中，最小值是 1 ，最大值是 4 。因此，2 或 3 都是有效答案。

示例 2：

输入：nums = [1,2]

输出：-1

解释：由于不存在既不是最大值也不是最小值的数字，我们无法选出满足题目给定条件的数字。因此，不存在答案，返回 -1 。

示例 3：

输入：nums = [2,1,3]

输出：2

解释：2 既不是最小值，也不是最大值，这个示例只有这一个有效答案。

提示：

1 <= nums.length <= 100

1 <= nums[i] <= 100

nums 中的所有数字互不相同

# 分析

方法一：模拟

思路：

可以通过以下步骤来实现：

1、找到数组中的最小值和最大值。

2、遍历数组，找到第一个既不是最小值也不是最大值的数字。

代码：

class Solution {

public:

int findNonMinOrMax(vector<int>& nums) {

int minVal = INT\_MAX, maxVal = INT\_MIN;

for (int num : nums) {

minVal = min(minVal, num);

maxVal = max(maxVal, num);

}

for (int num : nums) {

if (num != minVal && num != maxVal) {

return num;

}

}

return -1; // 如果没有找到符合条件的数字，返回 -1

}

};