5/5 Leetcode

题目：第三大的数

给你一个非空数组，返回此数组中 第三大的数 。如果不存在，则返回数组中最大的数。

示例 1：

输入：[3, 2, 1]

输出：1

解释：第三大的数是 1 。

示例 2：

输入：[1, 2]

输出：2

解释：第三大的数不存在, 所以返回最大的数 2 。

示例 3：

输入：[2, 2, 3, 1]

输出：1

解释：注意，要求返回第三大的数，是指在所有不同数字中排第三大的数。

此例中存在两个值为 2 的数，它们都排第二。在所有不同数字中排第三大的数为 1 。

来源：力扣（LeetCode）

链接：<https://leetcode-cn.com/problems/third-maximum-number>

思路：

第一个出现的想法：

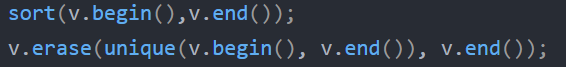
先从大到小排序，然后返回nums[2]就是第三大的数了，但是很快自己也发现这个是错了，如果nums的元素小于3会越界，如果出现一样的数，那nums[2]也未必是第三大的数。（因为自己想到了这个问题所以并没有写这个版本）

第二个出现的想法：（一版本）

还是先从大到小排序，然后如果nums元素个数小于3，则直接返回nums[0]，如果大于3，就判断当前数和下一个数是否相等，如果不等则flag+1，当flag==2时，就返回nums[i]就是第三大的数了。但是很快又遇到错误了，如果数组是[1,2,2]那么应该返回什么呢，这样会直接报错，原因：flag永远到不了二，会先越界，然后返回一个乱码，所以要对这个问题再次进行修改。

第三个出现的想法：（二版本）

先从大到小排序，还是一样如果nums元素个数小于3，则直接返回nums[0]，如果nums元素大于3则，删除重复元素（通过unique和erase函数代码如下：）



然后再返回nums[2]即可；

这道题是在错误中一步又一步的向“正确答案”靠近的，从一开始的写代码，然后发现问题，小错误，再进行修改，把可能遇到的情况一一枚举出来，然后再对症下药，慢慢把题目解决的。

总结：

这道题其实说难也不难，说简单也不简单。不简单是因为有很多小情况需要考虑到，需要在一次次错误中发现自己的问题并进行改变，但是整体思路上来看并不算很复杂，慢慢摸索总会向正确答案靠近的。写完这道题也慢慢的知道要怎么去排除错误，针对错误来进行修改程序！