**难度: 简单**

给定一个非空数组，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。要求算法时间复杂度必须是O(n)。

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，是指第三大且唯一出现的数。

存在两个值为2的数，它们都排第二。

来源：力扣（LeetCode）

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

著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。

**思路:**

1). 分别创建第一大,第二大,第三大的变量为最小值

2). 从第三大到第一大,一直比较

3). 比如:一开始比较第三大,然后用第三大与第二大比较,然后第二大与第三大比较

4). 并且一开始并且第三大的时候,比较的值,不能等于第二大以及第一大,要不然会导致第三大的数据变大

**注:** 三个变量的值必须为long,因为测试案例有一个是[1,2,-2147483648]会超过int类型

|  |
| --- |
| public int thirdMax(int[] nums) {  long top1 = Long.MIN\_VALUE;  long top2 = Long.MIN\_VALUE;  long top3 = Long.MIN\_VALUE;  for (int i = 0; i < nums.length; i++) {    if (nums[i] > top3 && nums[i] != top2 && nums[i] != top1) {  top3 = nums[i];  if (top2 < top3) {  top3 = top2;  top2 = nums[i];  if (top1 < top2) {  top2 = top1;  top1 = nums[i];  }  }  }  }  return (int)(top3 == Long.MIN\_VALUE ? top1 : top3);  } |