# 题目

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。

# 分析

class Solution {

public:

int thirdMax(vector<int>& nums) {

std::set<int> tmpNums(nums.begin(),nums.end());

auto it = tmpNums.end();

it--;

if(tmpNums.size() >= 3)

{

it--;

it--;

}

return \*it;

}

};