414. Third Maximum Number:

Given an integer array `nums`, Return *the **third distinct maximum** number in this array. If the third maximum does not exist, return the **maximum** number.*

Example 1:

```
Input: nums = [3,2,1]
Output: 1
Explanation:
The first distinct maximum is 3.
The second distinct maximum is 2.
The third distinct maximum is 1.
```

Example 2:

```
Input: nums = [1,2]
Output: 2
Explanation:
The first distinct maximum is 2.
The second distinct maximum is 1.
The third distinct maximum does not exist, so the maximum
(2) is returned instead.
```

Example 3:

```
Input: nums = [2,2,3,1]
Output: 1
Explanation:
The first distinct maximum is 3.
The second distinct maximum is 2 (both 2's are counted toge
ther since they have the same value).
The third distinct maximum is 1.
```

Constraints:

- `1 <= nums.length <= 104`

- `231 <= nums[i] <= 231 - 1`

Solution: Time = O(n log n) and space = O(n)

```
class Solution:
    def thirdMax(self, nums: List[int]) -> int:
        s = set(nums)
        lst = sorted(s, reverse=True)

        if len(lst) >= 3:
            return lst[2]
        else:
            return lst[0]
```

Optimal solution: Time = O() and Space = O()

```
class Solution:
    def thirdMax(self, nums: list[int]) -> int:
        first = second = third = None

        for num in nums:
            if num == first or num == second or num == third:
                continue

            if first is None or num > first:
                first, second, third = num, first, second
            elif second is None or num > second:
                second, third = num, second
            elif third is None or num > third:
                third = num

        return third if third is not None else first
```