414 Third Maximum Number

2018年4月11日 10:54

Given a **non-empty** array of integers, return the **third** maximum number in this array. If it does not exist, return the maximum number. The time complexity must be in O(n).

Example 1:

Input: [3, 2, 1]

Output: 1

Explanation: The third maximum is 1.

Example 2:

Input: [1, 2]

Output: 2

Explanation: The third maximum does not exist, so the maximum (2) is returned instead.

Example 3:

Input: [2, 2, 3, 1]

Output: 1

Explanation: Note that the third maximum here means the third maximum distinct number. Both numbers with value 2 are both considered as second maximum.

来自 <https://leetcode.com/problems/third-maximum-number/description/>

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)。

Solution for Python3:

```python
class Solution:
    def thirdMax(self, nums):
        """
        :type nums: List[int]
        :rtype: int
        """
        top3 = [float('-inf'), float('-inf'), float('-inf')]
        for num in nums:
           if num not in top3:
               if num > top3[0]:
                   top3 = [num, top3[0], top3[1]]
               elif num > top3[1]:
                   top3 = [top3[0], num, top3[1]]
               elif num > top3[2]:
                   top3 = [top3[0], top3[1], num]
        return max(nums) if float('-inf') in top3 else top3[2]

```

Solution for C++:

```cpp
class Solution1 {
public:
    int thirdMax(vector<int>& nums) {
        int f = INT_MIN, s = INT_MIN, t = INT_MIN; //#include<limits.h>
        int m = 0;
        int uniqCnt = 0;
```

```
7           for (int i = 0; i < nums.size(); i++) {
8               if (nums[i] == f || nums[i] == s || nums[i] == t) {
9                   if (nums[i] == INT_MIN)
10                      m = 1;
11                  continue;
12              }
13              if (nums[i] > f) {
14                  t = s;
15                  s = f;
16                  f = nums[i];
17              } else if (nums[i] > s) {
18                  t = s;
19                  s = nums[i];
20              } else if (nums[i] > t) {
21                  t = nums[i];
22              }
23              uniqCnt++;
24          }
25          return uniqCnt + m < 3 ? f : t;
26      }
27  };
28
29  class Solution2 {
30  public:
31      int thirdMax(vector<int>& nums) {
32          set<int> top3;
33          for (int num : nums) {
34              top3.insert(num);
35              if (top3.size() > 3) {
36                  top3.erase(top3.begin());
37              }
38          }
39          return top3.size() == 3 ? *top3.begin() : *top3.rbegin();
40      }
41  };
42
43  class Solution3 {
44  public:
45      int thirdMax(vector<int>& nums) {
46          set<int> top3;
47          for (int num : nums) {
48              if (top3.insert(num).second && top3.size() > 3) {
49                  top3.erase(top3.begin());
50              }
51          }
52          return top3.size() == 3 ? *top3.begin() : *top3.rbegin();
53      }
54  };
```