Second Largest in an array

Problem Description:
You are given an array of integers and you have to find the second largest element present in the given array.

Approach 1: The simplest solution of doing this problem is to sort the array in descending order and return the second element, or sort in ascending order and return the 2nd element from last. But, this won't work in case of duplicate elements.
Time Complexity: Time complexity of this approach is O(nlogn) for sorting an array.

Approach 2: In this approach, you can find the largest element present in the array. So, to find the largest element let us say largest is minus infinity initially, now scan the whole array once, if a particular element is greater than the largest then update the largest element to that element. Then make a second scan to find an element just smaller than the largest element.

Time complexity: Time complexity of this approach is O(n) for scanning n elements of an array. But instead of doing 2 scans, we can find the second largest element in just one scan only. To find the element in a single scan we will move to the next approach.

Approach 3: This is an efficient solution to find the second largest element in a single scan. In this approach we will maintain both maximum and second maximum elements with us at a time and will continue to update both by scanning the array only once.

Steps to proceed:

1. Initialize 2 variables as INT_MIN (minus infinity) initially.
   maxx = secmax = INT_MIN

maxx=minus_infinity *secmax=minus_infinity*

2. If there are less than 2 elements in the array then there will be no second largest, then just return INT_MIN.

```
If n is less than 2:
return minus_infinity
```

3. Start traversing the array,
   a) If the current element in array say arr[i] is greater than the maxx. Then update maxx and secmax as,
   secmax = maxx
   maxx = arr[i]
   b) If the current element is in between maxx and secmax, then update secmax to store the value of current variable as
   secmax = arr[i]

```
For i=0 to i less than n:
        If input[i]>maxx:
            secmax=maxx
            maxx=input[i]
        Else if input[i]> secmax and input[i] not equal to maxx:
            secmax=input[i]
```

4. Return the value stored in secmaxx.

```
Return secmax
```

Time Complexity: The time complexity of this approach is O(n) as we have to traverse the whole array once.

❏ Let us see this approach with an example :-
arr={1,5,3,6,7}

- maxx = secmax = INT_MIN
- Now, arr[0]>maxx, then maxx=arr[0]=1
  secmax=INT_MIN

- arr[1]>maxx, then secmax=maxx=1
  maxx=arr[1]=5
- secmax<arr[2]<maxx, then maxx=5
  secmax=3
- arr[3]>maxx, then secmax=maxx=5
  maxx=arr[3]=6
- arr[4]>maxx, then secmax=maxx=6
  maxx=arr[4]=7
- Return secmax=6 which is the 2nd largest element present in the array.