

Introduction to Arrays

(Introduction to Problem Solving I)

Additional Problems

Q5. Second Largest </>

✓ Solved



Using hints except Complete Solution is Penalty free now

[Use Hint](#)

Problem Description

You are given an integer array **A**. You have to find the second largest element/value in the array or report that no such element exists.

Problem Constraints

$$1 \leq |A| \leq 10^5$$

$$0 \leq A[i] \leq 10^9$$

Input Format

The first argument is an integer array **A**.

Output Format

Return the second largest element. If no such element exist then return -1.

Example Input

Input 1:

A = [2, 1, 2]

Input 2:

A = [2]

Example Input

Input 1:

A = [2, 1, 2]

Input 2:

A = [2]

Example Output

Output 1:

1

Output 2:

-1

Example Explanation

Explanation 1:

First largest element = 2

Second largest element = 1

Explanation 2:

There is no second largest element in the array.

18	17	1	3	5	16
----	----	---	---	---	----

①

max = 18

secondMax = 18 → this will never get updated

②

max = $-\infty$

secondMax = $-\infty$

i) 18
max = 18

if (ele > max) {

secondMax = max

max = ele

}

this way

we have the
previous max
stored

but second max = ~~$-\infty$~~

∴ we also need an else-if condition

```
else if (ele > secondMax) {
```

```
    secondMax = ele
```

```
}
```

⇒ but what if the array only one element

18

Initial

max = $-\infty$

sMax = $-\infty$

→ i) ele = 18

sMax = $-\infty$

max = 18

⇒ so for that we can have a check before the algorithm even starts

```
if (A.length == 1)
```

```
    return -2;
```

// Algorithm skeleton

```
if (A.length == 1) {
```

```
    return -2;
```

```
}
```

```
int max = Integer.MINIMUM-VALUE;
```

```
int secondMax = max;
```

} both are lowest possible integers

```
for (int i = 0; i < A.length; i++) {
```

```
    if (A[i] > max) {
```

```
        secondMax = max;
```

```
        max = A[i];
```

```
    } else if (A[i] > secondMax) {
```

```
        secondMax = A[i];
```

```
    }
```

```
}
```

NOT CORRECT!!!

See below for correction

return max > secondMax ? secondMax : -1

Wait!!!

but what if

18	5	13	18	17	16	17
----	---	----	----	----	----	----

max = 18
sMax = -∞

sMax = 13
sMax = 5

sMax = 18!
NO!!!

else it will make
sMax 18!

⇒ we need to modify else if

⇒ else if (A[i] > secondMax && A[i] < max)



Why? Observation

In the problem constraints it is mentioned

Problem Constraints

$$1 \leq |A| \leq 10^5$$

$$0 \leq A[i] \leq 10^9$$

elements of the array will always be greater than '0'

→ so?

→ you don't need to set max, secondMax as Integer.MIN-VALUE

→ You can just set it as -1

→ why?

→ We have to return the second largest element

→ If it doesn't exist, we anyways have to return -1

→ So, if the algorithm was unable to set any element as the second Maximum, anyways just returning second Maximum would be fine

// Skeleton (Hope it's right this time)

```
if (A.length == 1)
    return -1;
```

```
int max = -1;
```

```
int secondMax = -1;
```

```
for (int i = 0; i < A.length; i++) {
```

```
    if (A[i] > max) {
```

```
        secondMax = max;
```

```
        max = A[i];
```

```
    } else if (A[i] > secondMax && A[i] < max) {
```

```
        secondMax = A[i];
```

```
    }
```

```
}
```

```
return secondMax;
```



Accepted