Problem statement Send feedback

You have been given an array/list 'ARR' consisting of 'N' integers. Your task is to find the majority element in the array. If there is no majority element present, print -1.

Note:

A majority element is an element that occurs more than floor('N' / 2) times in the array.

Detailed explanation (Input/output format, Notes, Images)

Constraints:

1 <= T <= 100
1 <= N <= 5 * 10^3
-10^5 <= ARR[i] <= 10^5

Where 'ARR[i]' denotes the element at the 'i'th index in the array/list 'ARR'.

Time limit: 1 sec

Sample Input 1:

2
5
2 3 9 2 2
4
8 5 1 9

Sample Output 1:

2
-1

Explanation of Sample Output 1:

In test case 1, frequencies of occurrences of different elements are:

2 → 3 times
3 → 1 time
9 → 1 time

As 2 occurs more than floor(5/2) (i.e. floor(2.5) = 2) times, it is the majority element.

In test case 2, frequencies of occurrences of different elements are:

8 → 1 time
5 → 1 time
1 → 1 time
9 → 1 time

As no element occurs more than floor(4/2) = 2 times. Thus No majority element is present.

Sample Input 2:

```
2
7
8 8 8 8 8 9 1
4
2 2 3 3
```

Sample Output 2:

```
8
-1
```

Explanation of Sample Output 2:

```
In test case 1, frequencies of occurrences of different elements are:

8 → 5 times
9 → 1 time
1 → 1 time

As 8 occurs more than floor(7/2) (i.e. floor(3.5) = 3) times, it is the majority
element.

In test case 2, frequencies of occurrences of different elements are:

2 → 2 times
3 → 2 times

As no element occurs more than floor(4/2) = 2 times. Thus No majority element is
present.
```