Task description

A non-empty array A consisting of N integers and sorted in a non-decreasing order (i.e. A[0] ≤ A[1] ≤ ... ≤ A[N−1]) is given. The *leader* of this array is the value that occurs in more than half of the elements of A.

You are given an implementation of a function:

```
class Solution { public int solution(int[] A); }
```

that, given a non-empty array A consisting of N integers, sorted in a non-decreasing order, returns the leader of array A. The function should return −1 if array A does not contain a leader.

For example, given array A consisting of ten elements such that:

```
A[0] = 2 A[1] = 2 A[2] = 2 A[3] = 2 A[4] = 2 A[5] = 3 A[6] = 4 A[7] = 4
A[8] = 4 A[9] = 6
```

the function should return −1, because the value that occurs most frequently in the array, 2, occurs five times, and 5 is not more than half of 10.

Given array A consisting of five elements such that:

```
A[0] = 1 A[1] = 1 A[2] = 1 A[3] = 1 A[4] = 50
```

the function should return 1.

The attached code is still incorrect for some inputs. Despite the error(s), the code may produce a correct answer for the example test cases. The goal of the exercise is to find and fix the bug(s) in the implementation. You can modify at most three lines.

Assume that:

- N is an integer within the range [1..100,000];
- each element of array A is an integer within the range [0..2,147,483,647];
- array A is sorted in non-decreasing order.

In your solution, focus on correctness. The performance of your solution will not be the focus of the assessment.