Codility_

Candidate Report: trainingW9JXX7-6SP

Check out Codility training tasks

Test Name:

Summary Review (0) Timeline

Tasks summary

Task	Time spent	Score
Dominator Java 8	2 min	100%

Total score

Tasks Details

Easy

1. Dominator

Find an index of an array such that its value occurs at more than half of indices in the array.

Task Score	Correctness	Performance
100%	100%	100%

Task description

An array A consisting of N integers is given. The *dominator* of array A is the value that occurs in more than half of the elements of A.

For example, consider array A such that

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

The dominator of A is 3 because it occurs in 5 out of 8 elements of A (namely in those with indices 0, 2, 4, 6 and 7) and 5 is more than a half of 8.

Write a function

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

that, given an array A consisting of N integers, returns index of any element of array A in which the dominator of A occurs. The function should return −1 if array A does not have a dominator.

For example, given array A such that

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

the function may return 0, 2, 4, 6 or 7, as explained above.

Write an **efficient** algorithm for the following assumptions:

- N is an integer within the range [0..100,000];

Solution

Programming language used:	Java 8
Total time used:	2 minutes
Effective time used:	2 minutes
Notes:	*not defined yet*

Task timeline

22:33:12 22:34:37

Code: 22:34:37 UTC, java, final, score: **100**

show code in pop-up

```
// you can also use imports, for example:
// import java.util.*;

// you can write to stdout for debugging purposes, e.g.
// System.out.println("this is a debug message");
import java.util.Hashtable;
class Solution {
```

- each element of array A is an integer within the range [−2,147,483,648..2,147,483,647].

```
    public int solution(int[] A) {

                // write your code in Java SE 8
                Hashtable<Integer, Integer> occuranceTable

                if (A != null) {
                        int dominLimit = A.length / 2;
                        if (A.length ==1)
                                return 0;
                        else if(A.length>1){
                                for (int i = 0; i<A.length
                                        if (occuranceTable
                                                occuranceT
                                        } else {
                                                int count
                                                count++;
                                                if (count
                                                        re
                                                else
                                                        oc
                                        }
                                }

                        }
                }
                return -1;

    }
}
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity: **O(N*log(N)) or O(N)**

expand all Example tests

▶ example ✓ OK
example test

expand all Correctness tests

▶ small_nondominator ✓ OK
all different and all the same elements

▶ small_half_positions ✓ OK
half elements the same, and half + 1 elements the same

▶ small ✓ OK
small test

▶ small_pyramid ✓ OK
decreasing and plateau, small

▶ extreme_empty_and_single_item ✓ OK
empty and single element arrays

▶ extreme_half1 ✓ OK
array with exactly N/2 values 1, N even + [0,0,1,1,1]

▶ extreme_half2 ✓ OK
array with exactly floor(N/2) values 1, N odd + [0,0,1,1,1]

▶ extreme_half3 ✓ OK
array with exactly ceil(N/2) values 1 + [0,0,1,1,1]

Performance tests

expand all

▶	medium_pyramid decreasing and plateau, medium	✓ OK
▶	large_pyramid decreasing and plateau, large	✓ OK
▶	medium_random random test with dominator, N = 10,000	✓ OK
▶	large_random random test with dominator, N = 100,000	✓ OK

The PDF version of this report that may be downloaded on top of this site may contain sensitive data including personal information. For security purposes, we recommend you remove it from your system once reviewed.