Codility_

CodeCheck Report: trainingWETEYC-T7Z

Test Name:

Check out Codility training tasks

Summary Timeline

Tasks summary

Task	Time spent	Score
Dominator Python	3 min	100%

Total score

Tasks Details

1. **Dominator**

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

Easy

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
```

Solution

Programming language used:	Python
Total time used:	3 minutes
Effective time used:	3 minutes
Notes:	*not defined yet*

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

```
def solution(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

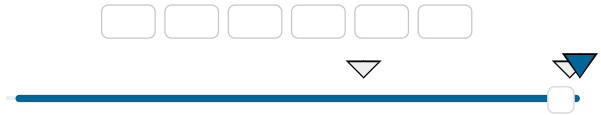
```
A[0] = 3    A[1] = 4    A[2] =  3
A[3] = 2    A[4] = 3    A[5] = −1
A[6] = 3    A[7] = 3
```

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

Task timeline



11:50:23 11:53:12

Code: 11:53:11 UTC, py, final, score: **100** show code in pop-up

```
1  # you can write to stdout for debuggin
2  # print("this is a debug message")
3
4  def solution(A):
5      # Implement your solution here
6      # pass
7      if len(A) == 0:
8          return -1
9
10     sort_a = sorted(A)
11     l = len(A) // 2
12     get_denominator = sort_a[l]
13     number_of_denominator = sort_a.cou
14     if number_of_denominator > l:
15         return A.index(get_denominator
16
17     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 example test	✔ OK
expand all	**Correctness tests**	
▶	small_nondominator all different and all the same elements	✔ OK
▶	small_half_positions half elements the same, and half + 1 elements the same	✔ OK
▶	small small test	✔ OK
▶	small_pyramid	✔ OK

decreasing and plateau, small		
▶	extreme_empty_and_single_item empty and single element arrays	✔ OK
▶	extreme_half1 array with exactly N/2 values 1, N even + [0,0,1,1,1]	✔ OK
▶	extreme_half2 array with exactly floor(N/2) values 1, N odd + [0,0,1,1,1]	✔ OK
▶	extreme_half3 array with exactly ceil(N/2) values 1 + [0,0,1,1,1]	✔ OK
expand all	**Performance tests**	
▶	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