Codility_

CodeCheck Report: trainingSHREMS-BKF

Test Name:

Check out Codility training tasks

Summary Timeline AI Assistant Transcript

Tasks summary

Task	Time spent	Score
Dominator C++ ⚠	8 min	100%

Total score

100%

Tasks Details

1. Dominator

Easy

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

```
int solution(vector<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
```

Solution

Programming language used:	C++	
Total time used:	8 minutes	?
Effective time used:	8 minutes	?
Notes:	*not defined yet*	

Task timeline

?

07:20:17 07:28:16

Code: 07:28:16 UTC, cpp, final, score: **100**

show code in pop-up

```
// you can use includes, for example:
// #include <algorithm>
```

```
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].

```

// you can write to stdout for debugging purpo
// cout << "this is a debug message" << endl;

int solution(vector<int> &A) {
    int c;

//맞춰볼 숫자가 있는 상황인지 확인하기 위해 size 선언
    int size = 0;
    for (int a : A) {
        if (size == 0) {
            c = a;
            size = 1;
        }
        else {
            if (c == a) {
                size++;
            } else {
                size--;
            }
        }
    }
//The dominator of array A is the value that
    int count = 0;
    int index;

    for (int i = 0; i < A.size(); i++) {
        if (A[i] == c) {
            count++;
            index = i;
        }
    }

    if (count > A.size()/2)
        return index;

    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 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