Codility_

Candidate Report: trainingBFRY8K-WNV

Check out Codility training tasks

Test Name:

Summary Review (0) Timeline

Tasks summary

Task	Time spent	Score
OddOccurrencesInArray Java 8	18 min	100%

Total score

100%

Tasks Details

Easy

1. OddOccurrencesInArray

Find value that occurs in odd number of elements.

Task Score	Correctness	Performance
100%	100%	100%

Task description

A non-empty array A consisting of N integers is given. The array contains an odd number of elements, and each element of the array can be paired with another element that has the same value, except for one element that is left unpaired.

For example, in array A such that:

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

- the elements at indexes 0 and 2 have value 9,
- the elements at indexes 1 and 3 have value 3,
- the elements at indexes 4 and 6 have value 9,
- the element at index 5 has value 7 and is unpaired.

Write a function:

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

that, given an array A consisting of N integers fulfilling the above conditions, returns the value of the unpaired element.

For example, given array A such that:

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

the function should return 7, as explained in the example above.

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

Solution

Programming language used:	Java 8
Total time used:	18 minutes
Effective time used:	18 minutes
Notes:	*not defined yet*

Task timeline

03:43:14 04:00:44

Code: 04:00:43 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.HashMap;
import java.util.Map;

```

- N is an odd integer within the range [1..1,000,000];
- each element of array A is an integer within the range [1..1,000,000,000];
- all but one of the values in A occur an even number of times.

```
class Solution {
    public int solution(int[] A) {
        // write your code in Java SE 8
                int result = 0, count = 0;

                Map<Integer,Integer> map = new HashMap<Int
                for(Integer i : A) {
                        if(map.containsKey(i))
                        {
                                count = (int)map.get(i);
                                map.remove(i);
                                map.put(i, ++count);
                        }else {
                                map.put(i, 1);
                        }
                }
                for(Integer i : map.keySet()) {
                        if((map.get(i)%2)==1) {
                                result = i;
                                break;
                        }
                }

                return result;

    }
}
```

Analysis summary

The solution obtained perfect score.

Analysis ?

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

expand all **Example tests**

▶ example1 ✓ OK
example test

expand all **Correctness tests**

▶ simple1 ✓ OK
simple test n=5

▶ simple2 ✓ OK
simple test n=11

▶ extreme_single_item ✓ OK
[42]

▶ small1 ✓ OK
small random test n=201

▶ small2 ✓ OK
small random test n=601

expand all **Performance tests**

▶ medium1 ✓ OK
medium random test n=2,001

▶ medium2 ✓ OK
medium random test n=100,003

▶ big1 ✓ OK
big random test n=999,999, multiple repetitions

▶ big2 ✓ OK
big random test n=999,999