Codility_

CodeCheck Report: trainingH3QKHA-9Y6

Test Name:

Check out Codility training tasks

Summary Timeline AI Assistant Transcript

Tasks summary

Task		Time spent	Score
OddOccurrencesInArray Java 8	⚠	1 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); }
```

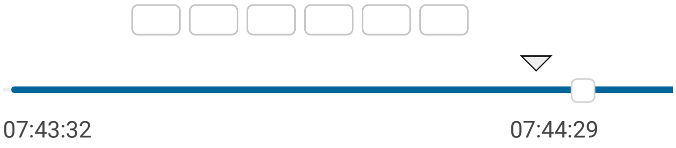
that, given an array A consisting of N integers fulfilling the above conditions, returns the value of the unpaired element.

For example, given array A such that:

Solution

Programming language used:	Java 8	
Total time used:	1 minutes	?
Effective time used:	1 minutes	?
Notes:	*not defined yet*	

Task timeline



Code: 07:44:28 UTC, java, final, score: **100**

show code in pop-up

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

```
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:

- 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.

```
import java.util.HashMap;
import java.util.Map;

// you can write to stdout for debugging purposes
// System.out.println("this is a debug message");

class Solution {
    public int solution(int[] A) {
        // Implement your solution here
        int OddOccuringElement = 0;

        Map<String, Integer> map = new HashMap<St
        for(int item : A) {
            Integer value = map.get(Integer.toStr
            if(value == null) {
                map.put(Integer.toString(item), i

            }
            else {
                map.remove(Integer.toString(item)
            }
        }
        OddOccuringElement=(int) map.values().toA
        //System.out.println(OddOccuringElement);

        return OddOccuringElement;
    }
}
```

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 big random test n=999,999	✓ OK