



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	Task Score	Correctness	Performance	
	Find value that occurs in odd number of elements.		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
```

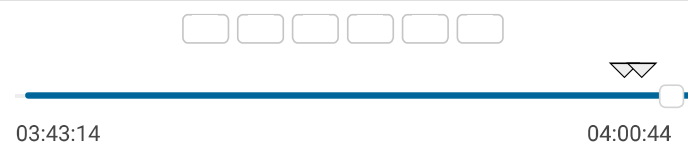
the function should return 7, as explained in the example above.

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



Code: 04:00:43 UTC, java, final, score: 100

[show code in pop-up](#)

```
1 // you can also use imports, for example:
2 // import java.util.*;
3
4 // you can write to stdout for debugging purposes, e.g.
5 // System.out.println("this is a debug message");
6 import java.util.HashMap;
7 import java.util.Map;
8
```

- 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.
- Copyright 2009–2020 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Test results - Codility

```
9  class Solution {
10     public int solution(int[] A) {
11         // write your code in Java SE 8
12         int result = 0, count = 0;
13
14         Map<Integer,Integer> map = new HashMap<Int
15         for(Integer i : A) {
16             if(map.containsKey(i))
17                 {
18                     count = (int)map.get(i);
19                     map.remove(i);
20                     map.put(i, ++count);
21                 }else {
22                     map.put(i, 1);
23                 }
24         }
25         for(Integer i : map.keySet()) {
26             if((map.get(i)%2)==1) {
27                 result = i;
28                 break;
29             }
30         }
31
32         return result;
33     }
34 }
35 }
```

Analysis summary

The solution obtained perfect score.

Analysis ?

Detected time complexity:

O(N) or
O(N*log(N))

expand all	Example tests
▶ example1 example test	✓ OK
expand all	Correctness tests
▶ simple1 simple test n=5	✓ OK
▶ simple2 simple test n=11	✓ OK
▶ extreme_single_item [42]	✓ OK
▶ small1 small random test n=201	✓ OK
▶ small2 small random test n=601	✓ OK
expand all	Performance tests
▶ medium1 medium random test n=2,001	✓ OK
▶ medium2 medium random test n=100,003	✓ OK
▶ big1 big random test n=999,999, multiple repetitions	✓ OK
▶ big2 big random test n=999,999	✓ OK