

Training ticket

Session

ID: training84FBKS-3RX
 Time limit: 120 min.

Status: closed

Created on: 2016-04-17 03:43 UTC
 Started on: 2016-04-17 03:43 UTC
 Finished on: 2016-04-17 04:01 UTC

Tasks in test

1 **OddOccurrencesInArray**
 Submitted in: Java

Correctness

100%

Performance

100%

Task score

100%

Test score

100%

100 out of 100 points

EASY

1. OddOccurrencesInArray

Find value that occurs in odd number of elements.

score: 100 of 100



Task description

A non-empty zero-indexed 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.

Assume that:

- 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.

Complexity:

- expected worst-case time complexity is O(N);
- expected worst-case space complexity is O(1), beyond input storage (not counting the storage required for input arguments).

Elements of input arrays can be modified.

Copyright 2009–2016 by Codility Limited. All Rights Reserved. Unauthorized copying,

Solution

Programming language used: Java

Total time used: 18 minutes

Effective time used: 19 minutes

Notes: not defined yet

Task timeline



03:43:55

04:01:56

Code: 04:01:56 UTC, java, final,
 score: 100

[show code in pop-up](#)

```
1  import java.util.Arrays;
2
3  class Solution {
4      public int solution(int[] A) {
5
6          Arrays.sort(A);
7
8          int count = 1;
9          int last = A[0];
10         for(int i = 1; i < A.length; i++) {
11             if(A[i] == last) {
12                 count++;
13             }
14             else if(count % 2 == 0) {
15                 count = 1;
16                 last = A[i];
17             }
18             else {
19                 return last;
20             }
21         }
22         return last;
23     }
24 }
```

Analysis summary

Analysis



Detected time complexity:
 $O(N)$ or $O(N \cdot \log(N))$

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		