

CodeCheck Report: trainingWU3MA7-A2A

[Check out Codility training tasks](#)

Test Name:

Summary    Timeline

Tasks summary

Task	Time spent	Score
OddOccurrencesInArray Java 8	2 min	100%

Total score



Tasks Details

	1. OddOccurrencesInArray	Task Score	Correctness	Performance
Easy	Find value that occurs in odd number of elements.		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.

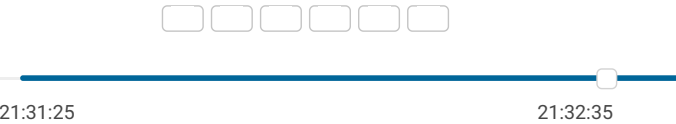
Write an **efficient** algorithm for the following assumptions:

- 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.

Solution

Programming language used:	Java 8
Total time used:	2 minutes ?
Effective time used:	2 minutes ?
Notes:	not defined yet

Task timeline ?



Code: 21:32:35 UTC, java, final, score: 100 [show code in pop-up](#)

```
1 import java.util.*;
2
3 class Solution {
4     public int solution(int[] A) {
5         Arrays.sort(A);
6         for (int i = 0; i < A.length - 1; i += 2) {
7             if (A[i] < A[i + 1]) {
8                 return A[i];
9             }
10        }
11        return A[A.length - 1];
12    }
13 }
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity:

$O(N)$  or  $O(N \cdot \log(N))$

collapse all	Example tests	
▼ example1	example test	✓ OK
1.	0.008 s	OK
collapse all	Correctness tests	
▼ simple1	simple test n=5	✓ OK
1.	0.008 s	OK
▼ simple2	simple test n=11	✓ OK
1.	0.004 s	OK
▼ extreme_single_item	[42]	✓ OK
1.	0.008 s	OK
▼ small1	small random test n=201	✓ OK
1.	0.008 s	OK
▼ small2	small random test n=601	✓ OK
1.	0.008 s	OK
collapse all	Performance tests	
▼ medium1	medium random test n=2,001	✓ OK
1.	0.008 s	OK
▼ medium2	medium random test n=100,003	✓ OK
1.	0.364 s	OK
▼ big1	big random test n=999,999, multiple repetitions	✓ OK
1.	2.560 s	OK
▼ big2	big random test n=999,999	✓ OK
1.	4.464 s	OK