codility

Check out Codility training tasks

Candidate Report: Anonymous

Test Name:

Summary Timeline

Test Score

100 out of 100 points

100%

Tasks in Test

	Time Spent	Task Score
OddOccurrencesInArray Submitted in: C++	17 min	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:

```
int solution(vector<int> &A);
```

Solution

Programming language used:	C++
Total time used:	17 minutes
Effective time used:	17 minutes
Notes:	*not defined yet*

Task timeline

11:28:13 11:44:35

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.

Code: 11:44:35 UTC, cpp, final, score: **100** show code in pop-up

```cpp
#include <vector>
#include <algorithm>

int solution(std::vector<int>& A)
{
   if (A.size() == 1) {
      return A[0];
   }

   std::sort(A.begin(), A.end());

   int result = A[A.size() - 1]; // Last element for
   for (size_t idx = 0; idx < A.size(); idx+=2) {
      if (A[idx] != A[idx + 1]) {
         result = A[idx];
         break;
      }
   }
   return result;
}
```

Analysis summary

The solution obtained perfect score.

Analysis

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

collapse all	Example tests	
▼ example1 example test		✓ OK
1. 0.001 s OK		
collapse all	Correctness tests	
▼ simple1 simple test n=5		✓ OK
1. 0.001 s OK		
▼ simple2 simple test n=11		✓ OK
1. 0.001 s OK		
▼ extreme_single_item [42]		✓ OK
1. 0.001 s OK		

▼ small1 ✓ OK
small random test n=201

1. 0.001 s OK

▼ small2 ✓ OK
small random test n=601

1. 0.001 s OK

collapse all Performance tests

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

1. 0.001 s OK

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

1. 0.012 s OK

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

1. 0.096 s OK

▼ big2 ✓ OK
big random test n=999,999

1. 0.160 s OK