Codility_

Candidate Report: training9QB4TC-CKF

Test Name:

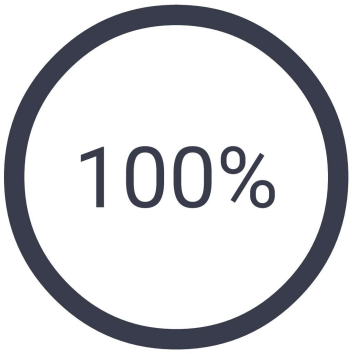
Check out Codility training tasks

Summary Timeline

Tasks summary

Task	Time spent	Score
OddOccurrencesInArray Python	31 min	100%

Total score



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:

```
def solution(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
```

Solution

Programming language used:	Python
Total time used:	31 minutes
Effective time used:	31 minutes
Notes:	*not defined yet*

Task timeline

03:03:42 03:34:08

Code: 03:34:08 UTC, py, final, score: **100** show code in pop-up

```
1  # you can write to stdout for debugging purposes,
2  # print("this is a debug message")
3
4  # <釐清問題>
```

```
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.


```python
#   找出一奇數長度陣列的整數，該整數在陣列中出現的次數
# <Solution>
#   排序 -> 透過slide window倆倆比對 -> 遇到不同的元
# <其他限制>
#       N is an odd integer within the range [1..1
#       each element of array A is an integer with
#       all but one of the values in A occur an ev

def solution(A):
    return findUnpairedNum(A)

def findUnpairedNum(array) -> int:
    sorted_array = sorted(array)

    sorted_array.append(0)
    sliding_times = int(len(sorted_array) / 2)

    for i in range(sliding_times):
        sliding_index = i*2
        s_window_head = sorted_array[sliding_index
        s_window_tail = sorted_array[sliding_index
        if not s_window_head == s_window_tail:
            return s_window_head
```

Analysis summary

The solution obtained perfect score.

Analysis

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

collapse all — Example tests

▼ example1 ✓ OK
example test

1. 0.036 s OK

collapse all — Correctness tests

▼ simple1 ✓ OK
simple test n=5

1. 0.036 s OK

▼ simple2 ✓ OK
simple test n=11

1. 0.036 s OK

▼ extreme_single_item ✓ OK
[42]

1. 0.036 s OK

▼ small1 ✓ OK
small random test n=201

1. 0.036 s OK

▼ small2 ✓ OK
small random test n=601

1. 0.036 s OK

collapse all Performance tests

▼	medium1 medium random test n=2,001	✓ OK
1.	0.040 s OK	
▼	medium2 medium random test n=100,003	✓ OK
1.	0.168 s OK	
▼	big1 big random test n=999,999, multiple repetitions	✓ OK
1.	1.384 s OK	
▼	big2 big random test n=999,999	✓ OK
1.	1.644 s OK	