codility

g center

lity training tasks

Congratulations

You have completed a Codility training test.

Tweet this!

I scored 100% in #python on @Codility!

https://codility.com/demo/take-sample-test/odd_occurrence

Sign up for our newsletter!

Like us on Facebook!

Training ticket

Session

ID: trainingG27NZF-VR3
Time limit: 120 min.

Status: closed

Created on: 2017-08-31 15:21 UTC
Started on: 2017-08-31 15:21 UTC
Finished on: 2017-08-31 16:18 UTC

Tasks in test	Correctness	Performance	Task score
1 | OddOccurrencesInArray Submitted in: Python	100%	100%	100%

Test score ?

100%

100 out of 100 points

How likely are you to recommend Codility to your friends and colleagues?

×

Not at all likely *Extremely likely*

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:

```
    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
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:

Solution

Programming language used: Python

Total time used: 58 minutes

Effective time used: 58 minutes

Notes: *not defined yet*

Task timeline

15:21:23 16:18:55

Code: 16:18:55 UTC, py, final, score: **100**

show code in pop-up

```
def solution(array):
  non_matches_hash = {}

  array_length = len(array)
  for i in xrange(array_length):
    val = array[i]

    existing_value = non_matches_hash.get(val)
    if existing_value == None:
      non_matches_hash[val] = 1
    else:
      # remove val from hash as there is a match now
      del non_matches_hash[val]

  return non_matches_hash.keys()[0]
```

Analysis summary

- 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.

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

Training center