Open Feedback Dialog

codility

Training center

Check out Codility training tasks

# Demo ticket

## Session

**ID:** demo76WVAX-3KB
**Time limit:** 120 min.

## Status: closed

**Created on:** 2014-03-15 04:34 UTC
**Started on:** 2014-03-15 04:34 UTC
**Finished on:** 2014-03-15 04:35 UTC

## Tasks in test

## Task score

## Test score

?

100%

100 out of 100 points

EASY

### 1. Dominator

Find an index of an array such that its value occurs at more than half of indices in the array.

**score: 100 of 100**

### Task description

A zero-indexed array A consisting of N integers is given. The *dominator* of array A is the value that occurs in more than half of the elements of A.
For example, consider array A such that

```
A[0] = 3    A[1] = 4    A[2] =  3
A[3] = 2    A[4] = 3    A[5] = −1
A[6] = 3    A[7] = 3
```

The dominator of A is 3 because it occurs in 5 out of 8 elements of A (namely in those with indices 0, 2, 4, 6 and 7) and 5 is more than a half of 8.
Write a function

```
def solution(A)
```

that, given a zero-indexed array A consisting of N integers, returns index of any element of array A in which the dominator of A occurs. The function should return −1 if array A does not have a dominator.
Assume that:

- N is an integer within the range [0..100,000];
- each element of array A is an integer within the range [−2,147,483,648..2,147,483,647].

For example, given array A such that

```
A[0] = 3    A[1] = 4    A[2] =  3
A[3] = 2    A[4] = 3    A[5] = −1
A[6] = 3    A[7] = 3
```

the function may return 0, 2, 4, 6 or 7, as explained above.
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.

### Solution

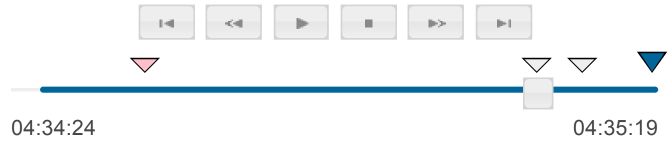
**Programming language used:** Python

**Total time used: 1 minutes** ?

**Effective time used: 1 minutes** ?

**Notes:** correct functionality and scalability

**Task timeline** ?



Code: 04:35:09 UTC, py, verify, result: **Passed**

```
def solution(A):
    # write your code in Python 2.6
    if not A:
        return -1

    candidate = A[0]
    count = 1
    size = len(A)
    for i in xrange(1, size):
        if A[i] == candidate:
            count += 1
        elif count > 0:
            count -= 1
        else:
            candidate = A[i]
            count = 1

    # verify candidate
    count = 0
    occur = -1
    for i in xrange(size):
        if A[i] == candidate:
            count += 1
            occur = i
    return occur if count > size/2 else -1
```

## Analysis

| test | time | result |
|---|---|---|
| example<br>example test | 0.050 s. | **OK** |

Training center