

Demo ticket

Session

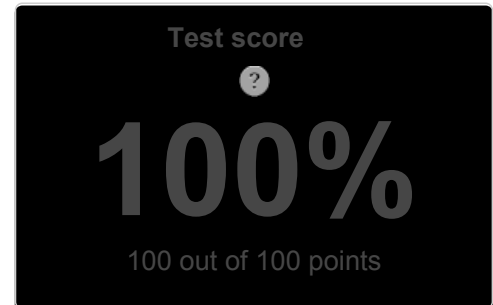
ID: demoZAJE46-W4W
Time limit: 120 min.

Status: closed

Created on: 2014-03-17 19:16 UTC
Started on: 2014-03-17 19:16 UTC
Finished on: 2014-03-17 19:18 UTC

Tasks in test

Task score



EASY

1. PassingCars

Count the number of passing cars on the road.

score: 100 of 100



Task description

A non-empty zero-indexed array A consisting of N integers is given. The consecutive elements of array A represent consecutive cars on a road.

Array A contains only 0s and/or 1s:

- 0 represents a car traveling east,
- 1 represents a car traveling west.

The goal is to count passing cars. We say that a pair of cars (P, Q), where $0 \leq P < Q < N$, is passing when P is traveling to the east and Q is traveling to the west.

For example, consider array A such that:

```
A[0] = 0
A[1] = 1
A[2] = 0
```

```
A[3] = 1
A[4] = 1
```

We have five pairs of passing cars: (0, 1), (0, 3), (0, 4), (2, 3), (2, 4). Write a function:

```
def solution(A)
```

that, given a non-empty zero-indexed array A of N integers, returns the number of passing cars.

The function should return -1 if the number of passing cars exceeds 1,000,000,000.

For example, given:

```
A[0] = 0
A[1] = 1
A[2] = 0
A[3] = 1
A[4] = 1
```

the function should return 5, as explained above. Assume that:

- N is an integer within the range [1..100,000];

Solution

Programming language used: Python

Total time used: 2 minutes

Effective time used: 2 minutes

Notes: correct functionality and scalability

Task timeline



Code: 19:18:19 UTC, py, final, score: 100.00

```
01. def solution(A):
02.     ones = 0
03.     ret = 0
04.     for i in xrange(len(A)-1, -1, -1):
05.         if A[i] == 1:
06.             ones += 1
07.         else:
08.             ret += ones
09.             if ret > 1000000000:
10.                 return -1
11.     return ret
```

Analysis

Detected time complexity:

O(N)

test	time	result
example		

- each element of array A is an integer within the range [0..1].

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.

Copyright 2009–2014 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Codility		
example	0.050 s.	OK
example test		
single	0.050 s.	OK
single element		
double	0.050 s.	OK
two elements		
simple	0.050 s.	OK
simple test		
small_random	0.050 s.	OK
random, length = 100		
medium_random	0.070 s.	OK
random, length = ~10,000		
large_random	0.220 s.	OK
random, length = ~100,000		
large_big_answer	0.170 s.	OK
0..01..1, length = ~100,000		
large_alternate	0.200 s.	OK
0101..01, length = ~100,000		
large_extreme	0.220 s.	OK
large test with all 1s/0s, length = ~100,000		

Training center