Open Feedback Dialog

codility

Training center

Check out Codility training tasks

Congratulations

You have completed a Codility demo.

Tweet this!
I scored 100% in #py on @Codility!
http://codility.com/demo/take-sample-test/passing_cars/

Sign up for our newsletter!

Like us on Facebook!

Demo ticket

Session

ID: demoP6BMP4-2FX
Time limit: 120 min.

Status: closed

Created on: 2014-04-25 17:04 UTC
Started on: 2014-04-25 17:04 UTC
Finished on: 2014-04-25 17:12 UTC

Tasks in test	Correctness	Performance	Task score
1 | {} PassingCars	100%	100%	100%

Test score

?

100%

100 out of 100 points

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 ≤ 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
```

Solution

Programming language used: Python

Total time used: 9 minutes ?

Effective time used: 9 minutes ?

Notes: correct functionality and scalability

Task timeline ?

17:04:23 17:12:51

Code: 17:12:51 UTC, py, final, score: **100.00**

```
def solution(A):
    ne = 0
    nw = 0
    np = 0

    for v in A:
        if v == 0:
            ne += 1
        elif v == 1:
            nw += 1
            np += ne
            if np > 1000000000:
                return -1
        # invalid input
```

```
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];
- 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.

```
        else:
            return -1

    return np
```

Analysis

Detected time complexity:

O(N)

test	time	result
Example tests		
example example test	0.050 s.	**OK**
Correctness tests		
single single element	0.050 s.	**OK**
double two elements	0.050 s.	**OK**
simple simple test	0.050 s.	**OK**
small_random random, length = 100	0.050 s.	**OK**
Performance tests		
medium_random random, length = ~10,000	0.060 s.	**OK**
large_random random, length = ~100,000	0.220 s.	**OK**
large_big_answer 0..01..1, length = ~100,000	0.170 s.	**OK**
large_alternate 0101..01, length = ~100,000	0.200 s.	**OK**
large_extreme large test with all 1s/0s, length = ~100,000	0.200 s.	**OK**

Training center