Congratulations

You have completed a Codility training test.

Sign up for our newsletter!

Like us on Facebook!

Training ticket

Session

ID: trainingEF45YZ-JFM
Time limit: 120 min.

Status: closed

Created on: 2016-06-09 13:41 UTC
Started on: 2016-06-09 13:41 UTC
Finished on: 2016-06-09 13:42 UTC

Tasks in test	Correctness	Performance	Task score
1 | PassingCars Submitted in: Java	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:

```
class Solution { public int solution(int[] A); }
```

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

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

For example, given:

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

Solution

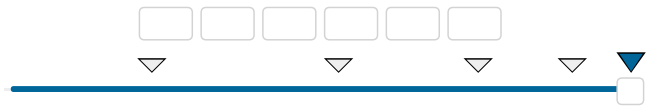
Programming language used: Java

Total time used: 1 minutes

Effective time used: 1 minutes

Notes: *not defined yet*

Task timeline



13:41:51 13:42:44

Code: 13:42:44 UTC, java, final, score: **100**

show code in pop-up

```
1  // you can also use imports, for example:
2  // import java.util.*;
3
4  // you can write to stdout for debugging purposes, e.g.
5  // System.out.println("this is a debug message");
6
7  class Solution {
8    public int solution(int[] a) {
9      int counter = 0;
10     int zeros = 0;
11     for (int i = 0; i < a.length; i++) {
12       if (a[i] == 0) {
13         zeros++;
14       } else if (a[i] == 1) {
15         counter += zeros;
```

```
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 that can have one of the following values: 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.


```
16        }
17        if (counter > 1_000_000_000) {
18          return -1;
19        }
20      }
21      return counter;
22    }
23  }
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity:
O(N)

expand all	Example tests	
▶	example example test	✔ OK
expand all	Correctness tests	
▶	single single element	✔ OK
▶	double two elements	✔ OK
▶	simple simple test	✔ OK
▶	small_random random, length = 100	✔ OK
▶	small_random2 random, length = 1000	✔ OK
expand all	Performance tests	
▶	medium_random random, length = ~10,000	✔ OK
▶	large_random random, length = ~100,000	✔ OK
▶	large_big_answer 0..01..1, length = ~100,000	✔ OK
▶	large_alternate 0101..01, length = ~100,000	✔ OK
▶	large_extreme large test with all 1s/0s, length = ~100,000	✔ OK