

Summary

Timeline

Tasks summary

Task	Time spent	Score
PassingCars Java 8	19 min	100%

Total score



Tasks Details

Easy	1. <b>PassingCars</b> Count the number of passing cars on the road.	Task Score	Correctness	Performance
		100%	100%	100%

Task description

A non-empty 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:

Write a function:

```
class Solution {
```

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

Solution

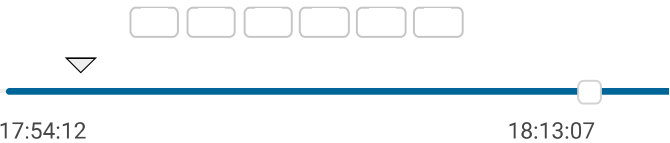
Programming language used: Java 8

Total time used: 19 minutes ?

Effective time used: 19 minutes ?

Notes: not defined yet

Task timeline ?



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

012345678910

Not at all likelyExtremely likely

show code in pop-up

for example:

```
for debugging purposes,
a debug message");
```

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

the function should return 5, as explained above.

Write an **efficient** algorithm for the following assumptions:

- 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.

Copyright 2009–2021 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

```
7
8  class Solution {
9      public int solution(int[] A) {
10         final int MAX_RESULT = 1000000000;
11         int result = 0, one_counter=0;
12         for(int i=A.length-1; i>=0; i--) {
13             if(A[i]==1) {
14                 one_counter++;
15             }
16             else {
17                 if(result>MAX_RESULT) {
18                     return -1;
19                 }
20                 result+=one_counter;
21             }
22         }
23         return result;
24     }
25 }
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity: **O(N)**

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

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

0

1

2

3

4

5

6

7

8

9

10

Not at all likely

Extremely likely