

Tasks summary

Task	Time spent	Score
PassingCars Python	2 min	100%

Total score

100%

Tasks Details

Easy

1. PassingCars

Count the number of passing cars on the road.

Task Score

100%

Correctness

100%

Performance

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: (0, 1), (0, 3), (0, 4), (2, 3), (2, 4).

Write a function:

```
def solution(A)
```

that, given a non-empty 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
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.

Solution

Programming language used: Python

Total time used: 2 minutes



Effective time used: 2 minutes



Notes: not defined yet

Task timeline



03:42:46

03:43:50

Code: 03:43:49 UTC, py, final, score: 100

[show code in pop-up](#)

```
1 # you can write to stdout for debugging purposes, e.g.
2 # print("this is a debug message")
3
4 def solution(A):
5     # write your code in Python 3.6
6     west = sum(A)
7     if west == 0 or west == len(A):
8         return 0
9     east = 0
10    npairs = 0
11    for el in A:
12        if el == 0:
13            east += 1
14            npairs += west
15            if npairs > 1000000000:
16                return -1
17        else:
18            west -= 1
19            if west == 0:
20                return npairs
21    return npairs
```

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_alterate 0101..01, length = ~100,000	✓ OK
▶ large_extreme large test with all 1s/0s, length = ~100,000	✓ OK