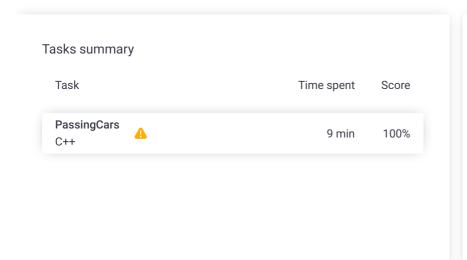
Codility_

CodeCheck Report: training52DUQP-A7K

Test Name:

Check out Codility training tasks

Summary Timeline 🛕 Al Assistant Transcript





Tasks Details

1. PassingCars Count the number of passing cars on the road.

Task Score Correctness Performance

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 $0 \le P < Q < N$, 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:

int solution(vector<int> &A);

Solution

Programming language used: C++

Total time used: 9 minutes

Effective time used: 9 minutes

Notes: not defined yet

Task timeline

4

15:03:59 15:12:45

Code: 15:12:45 UTC, cpp, show code in pop-up final, score: 100

- 1 // you can use includes, for example:
- 2 // #include <algorithm>
 3
 - // you can write to stdout for debugging purpo

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.

Copyright 2009–2024 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

```
Test results - Codility
```

```
// cout << "this is a debug message" << endl;</pre>
 7
     int solution(vector<int> &A) {
 8
         int N = A.size();
9
10
         int west = 0;
11
         int pass = 0;
12
         for (int i = N-1; i >= 0; i--) {
13
14
             if (A[i] == 1)
                  west++;
15
16
             if (A[i] == 0)
                 pass += west;
17
18
         if (pass > 1000000000)
19
             return -1;
20
21
22
         return pass;
23
     }
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity: O(N)

expa	and all Exan	ple tests
•	example example test	∨ OK
expand all Correctnes		tness 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
expa	and all Perform	nance tests
•	medium_random random, length = ~10,000	∨ OK
•	large_random random, length = ~100,000	∨ OK
•	large_big_answer 0011, length = ~100,000	∨ OK
•	large_alternate 010101, length = ~100,000	∨ OK
•	large_extreme large test with all 1s/0s, lengt ~100,000	✓ 0K h =