Get account with free plan

See how Codility works from recruiter's point of view.

closed

Demo ticket

Session
ID: demoSCXZQE-76U
Time limit: 120 min.

Status: closed
Started on: 2014-01-05 14:55 UTC

Score:

100

of 100

☆ 1. PassingCars

score: 100 of 100

Count the number of passing cars on the road.

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 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
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.

Solution

Programming language used: C#

Total time used: 5 minutes (?)

Effective time used: 1 minutes (?)

Notes: correct functionality and scalability

Task timeline What is it?

14:55:42 14:59:51

Code: 14:59:51 UTC, cs, final, score: 100.00

```
using System;
// you can also use other imports, for example:
// using System.Collections.Generic;
class Solution {
        public int solution(int[] A)
        {
            // write your code in C# with .NET 2.0
            var length = A.Length;
            if (length < 0 || length > 1000000) throw new ArgumentOutOfRangeException();

            var eastBoundCars = 0;
            var passingPairs = 0;

            foreach (var value in A)
            {
                if (value == 1)
                {
                    passingPairs += eastBoundCars;
                    if (passingPairs > 1000000000) return -1;
                }
                else
                    eastBoundCars++;
            }
            return passingPairs;
        }
}
```

Analysis

Detected time complexity:

O(N)

test	time	result
example example test	0.080 s.	**OK**
single single element	0.080 s.	**OK**
double two elements	0.080 s.	**OK**
simple simple test	0.080 s.	**OK**
small_random random, length = 100	0.080 s.	**OK**
medium_random random, length = ~10,000	0.080 s.	**OK**
large_random random, length = ~100,000	0.090 s.	**OK**
large_big_answer 0..01..1, length = ~100,000	0.090 s.	**OK**
large_alternate 0101..01, length = ~100,000	0.090 s.	**OK**
large_extreme large test with all 1s/0s, length = ~100,000	0.100 s.	**OK**

Get acco