

closed

Anders Persson

Session
ID: demo5M3DE6-C5S
Time limit: 30 min.

Status: closed
Started on: 2014-02-07 13:47 UTC

Score:

50

of 100

☆☆ 1. Equi

Find an index in an array such that its prefix sum equals its suffix sum.

score: 50 of 100 **Task description**

This is a demo task. You can read about this task and its solutions in this blog post.
A zero-indexed array A consisting of N integers is given. An *equilibrium index* of this array is any integer P such that $0 \leq P < N$ and the sum of elements of lower indices is equal to the sum of elements of higher indices, i.e.

$$A[0] + A[1] + \dots + A[P-1] = A[P+1] + \dots + A[N-2] + A[N-1].$$

Sum of zero elements is assumed to be equal to 0. This can happen if $P = 0$ or if $P = N-1$.
For example, consider the following array A consisting of $N = 7$ elements:

$$\begin{array}{lll} A[0] = -7 & A[1] = 1 & A[2] = 5 \\ A[3] = 2 & A[4] = -4 & A[5] = 3 \\ A[6] = 0 & & \end{array}$$

$P = 3$ is an equilibrium index of this array, because:

- $A[0] + A[1] + A[2] = A[4] + A[5] + A[6]$

$P = 6$ is also an equilibrium index, because:

- $A[0] + A[1] + A[2] + A[3] + A[4] + A[5] = 0$

and there are no elements with indices greater than 6.
 $P = 7$ is not an equilibrium index, because it does not fulfill the condition $0 \leq P < N$.

Write a function

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

that, given a zero-indexed array A consisting of N integers, returns any of its equilibrium indices. The function should return -1 if no equilibrium index exists.

Assume that:

- N is an integer within the range $[0..10,000,000]$;
- each element of array A is an integer within the range $[-2,147,483,648..2,147,483,647]$.

For example, given array A such that

$$\begin{array}{lll} A[0] = -7 & A[1] = 1 & A[2] = 5 \\ A[3] = 2 & A[4] = -4 & A[5] = 3 \\ A[6] = 0 & & \end{array}$$

the function may return 3 or 6, as explained above.
Complexity:

- expected worst-case time complexity is $O(N)$;
- expected worst-case space complexity is $O(N)$, beyond input storage (not counting the storage required for input arguments).

Elements of input arrays can be modified.

Copyright 2009–2014 by Codility Limited. All Rights Reserved.
Unauthorized copying, publication or disclosure prohibited.

Solution

Programming language used: Java

Total time used: 23 minutes

Effective time used: 23 minutes

Notes: not defined yet

Task timeline

13:47:46

14:10:14

Code: 14:10:14 UTC, java, final, score: 50.00

```
01. // you can also use imports, for example:
02. // import java.math.*;
03. class Solution {
04.     public int solution(int[] A) {
05.         // write your code in Java SE 6
06.
07.         int i = 0;
08.         int beforesum = 0;
09.         int beforeindex = 0;
10.         int aftersum = 0;
11.         int afterindex = 0;
12.         int result = -1;
13.         while ( i < A.length )
14.         {
15.             beforeindex = 0;
16.             aftersum = 0;
17.             beforesum = 0;
18.             while ( beforeindex < i )
19.             {
20.                 beforesum = beforesum+
21.                     A[beforeindex];
22.                 beforeindex = beforeindex
23.                     +1;
24.             }
25.             afterindex = i +1;
26.             while ( afterindex < A.length )
27.             {
28.                 aftersum = aftersum+
29.                     A[afterindex];
30.                 afterindex = afterindex +1;
31.             }
32.             if (aftersum == beforesum)
33.             {
34.                 result = i;
35.             }
36.             i = i + 1;
37.         }
38.         return result;
39.     }
```

Analysis

test	time	result
example Test from the task description	0.300 s.	OK
simple	0.300 s.	OK
extreme_large_numbers Sequence with extremely large numbers testing arithmetic overflow.	0.300 s.	WRONG ANSWER got 2, but it is not equilibrium point, sum[0..1]=429496 sum[3..3]=-2
extreme_negative_numbers Sequence with extremely large numbers testing arithmetic overflow.	0.290 s.	WRONG ANSWER got 2, but it is not equilibrium point, sum[0..1]=-42949 right sum (empty s
overflow_tests1 arithmetic overflow tests	0.300 s.	WRONG ANSWER got 0, but it is not equilibrium point, l sum (empty set)=(sum[1..2]=-42949
overflow_tests2 arithmetic overflow tests	0.290 s.	WRONG ANSWER got 2, but it is not equilibrium point, sum[0..1]=-42949 right sum (empty s
one_large one large number at the end of the sequence	0.290 s.	OK
sum_0 sequence with sum=0	0.290 s.	OK
single single number	0.300 s.	OK
empty Empty array	0.290 s.	OK
combinations_of_two multiple runs, all combinations of $\{-1,0,1\}^2$	0.290 s.	OK
combinations_of_three multiple runs, all combinations of $\{-1,0,1\}^3$	0.300 s.	OK
small_pyramid	0.310 s.	OK
large_long_sequence_of_ones	1.140 s.	TIMEOUT ERROR running time: >1.1 time limit: 0.98 sec
large_long_sequence_of_minus_ones	1.150 s.	TIMEOUT ERROR running time: >1.1 time limit: 0.98 sec
medium_pyramid	1.190 s.	TIMEOUT ERROR running time: >1.1 time limit: 0.94 sec
large_pyramid Large performance test, $O(n^2)$ solutions should fail.	2.190 s.	TIMEOUT ERROR running time: >2.1 time limit: 1.06 sec

Get acco