

AVAILABLE
LESSONS:*Lesson 1*
Iterations*Lesson 2*
Arrays*Lesson 3*
Time Complexity*Lesson 4*
Counting
Elements*Lesson 5*
Prefix Sums*Lesson 6*
Sorting*Lesson 7*
Stacks and
Queues*Lesson 8*
Leader*Lesson 9*
Maximum slice
problem*Lesson 10*
Prime and
composite
numbers*Lesson 11*

RESPECTABLE

MinAvgTwoSlice

START

Find the minimal average of any slice containing at least two elements.

Programming language: C++ ▼

A non-empty zero-indexed array A consisting of N integers is given. A pair of integers (P, Q) , such that $0 \leq P < Q < N$, is called a *slice* of array A (notice that the slice contains at least two elements). The *average* of a slice (P, Q) is the sum of $A[P] + A[P + 1] + \dots + A[Q]$ divided by the length of the slice. To be precise, the average equals $(A[P] + A[P + 1] + \dots + A[Q]) / (Q - P + 1)$.

For example, array A such that:
$$\begin{aligned} A[0] &= 4 \\ A[1] &= 2 \\ A[2] &= 2 \\ A[3] &= 5 \\ A[4] &= 1 \\ A[5] &= 5 \\ A[6] &= 8 \end{aligned}$$

contains the following example slices:

- slice $(1, 2)$, whose average is $(2 + 2) / 2 = 2$;
- slice $(3, 4)$, whose average is $(5 + 1) / 2 = 3$;
- slice $(1, 4)$, whose average is $(2 + 2 + 5 + 1) / 4 = 2.5$.

The goal is to find the starting position of a slice whose average is minimal.

Write a function:

```
int solution(vector<int> &A);
```

that, given a non-empty zero-indexed array A consisting of N integers, returns the starting position of the slice with the minimal average. If there is more than one slice with a minimal average, you should return the smallest starting position of such a slice.

Sieve of
Eratosthenes

Lesson 12

Euclidean
algorithm

Lesson 13

Fibonacci
numbers

Lesson 14

Binary search
algorithm

Lesson 15

Caterpillar
method

Lesson 16

Greedy
algorithms

Lesson 17

Dynamic
programming

Lesson 90

Tasks from
Indeed Prime
2016 challenge

Lesson 99

Future training

For example, given array A such that:

$A[0] = 4$

$A[1] = 2$

$A[2] = 2$

$A[3] = 5$

$A[4] = 1$

$A[5] = 5$

$A[6] = 8$

the function should return 1, as explained above.

Assume that:

- N is an integer within the range $[2..100,000]$;
- each element of array A is an integer within the range $[-10,000..10,000]$.

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 2009–2016 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

For programmers

[Lessons](#) [Challenges](#)
[Terms](#) [FAQ](#)

For companies

[Tour](#) [About us](#)
[Pricing](#) [Jobs](#)
[Blog](#) [Terms](#)
[Privacy](#) [Cookies](#)
[API](#)

Sign up for our newsletter:

Information about upcoming challenges, solutions and lessons directly in your inbox.

[Sign up](#)

Social:

[f](#) [t](#) [in](#)

Contact us:

For customer support queries:

UK +44 (0) 208 970 78 68

US 1-415-466-8085
support@codility.com

For sales queries:

UK +44 (0) 208 970 78 67

US 1-415-466-8085
sales@codility.com

© 2009–2016 Codility Ltd., registered in England and Wales (No. 7048726). VAT ID GB981191408. Registered office: 107 Cheapside, London EC2V 6DN