

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*

PAINLESS

## EquiLeader

START

Find the index  $S$  such that the leaders of the sequences  $A[0], A[1], \dots, A[S]$  and  $A[S + 1], A[S + 2], \dots, A[N - 1]$  are the same.

Programming language: C++ ▼

A non-empty zero-indexed array  $A$  consisting of  $N$  integers is given.

The *leader* of this array is the value that occurs in more than half of the elements of  $A$ .

An *equi leader* is an index  $S$  such that  $0 \leq S < N - 1$  and two sequences  $A[0], A[1], \dots, A[S]$  and  $A[S + 1], A[S + 2], \dots, A[N - 1]$  have leaders of the same value.

For example, given array  $A$  such that:

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

we can find two equi leaders:

- 0, because sequences: (4) and (3, 4, 4, 2) have the same leader, whose value is 4.
- 2, because sequences: (4, 3, 4) and (4, 4, 2) have the same leader, whose value is 4.

The goal is to count the number of equi leaders.

Write a function:

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

that, given a non-empty zero-indexed array  $A$  consisting of  $N$  integers, returns the number of equi leaders.

For example, given:

$A[0] = 4$

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

$A[1] = 3$

$A[2] = 4$

$A[3] = 4$

$A[4] = 4$

$A[5] = 2$

the function should return 2, 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  $[-1,000,000,000..1,000,000,000]$ .

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  
Pricing us  
Blog Jobs  
Privacy Terms  
Cookies  
API

**Sign up for our  
newsletter:**

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

**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