

Test Name:

Summary Timeline

Tasks summary

Task	Time spent	Score
EquiLeader C#	27 min	100%

Total score

100%

Tasks Details

Easy

1. EquiLeader

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

Task Score

100%

Correctness

100%

Performance

100%

Task description

A non-empty 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], ..., A[S] and A[S + 1], A[S + 2], ..., 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:

Solution

Programming language used: C#

Total time used:

27 minutes

?

Effective time used:

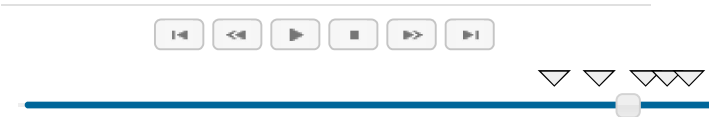
27 minutes

?

Notes:

not defined yet

Task timeline



- 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:

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

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

For example, given:

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

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

Copyright 2009–2022 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

12:13:19

12:39:28

Code: 12:39:27 UTC, cs, final,
score: 100

[show code in pop-up](#)

```
1  using System;
2  using System.Collections.Generic;
3
4  /**
5   * 8.2 - Equi Leader
6   * Paulo Santos
7   * 09.Dec.2022
8   */
9  class Solution {
10     public int solution(int[] A) {
11
12         var len = A.Length;
13         var dic = new Dictionary<int, int>();
14         var cnt = 0;
15         var lead = 0;
16         var equiLeader = 0;
17
18         foreach(var i in A) {
19             if (dic.ContainsKey(i)) {
20                 dic[i]++;
21                 if (dic[i] > cnt) {
22                     cnt = dic[i];
23                     lead = i;
24                 }
25             }
26             else
27                 dic[i] = 1;
28         }
29
30         if (cnt <= (len / 2))
31             return 0;
32
33         var lft = 0;
34         var rgt = cnt;
35         for (var i = 0; i < len; i++) {
36             if (A[i] == lead) {
37                 lft += 1;
38                 rgt -= 1;
39             }
40             if ((lft > ((i + 1) / 2)) &&
41                 (rgt > ((len - i - 1) / 2))) {
42                 equiLeader += 1;
43             }
44         }
45
46         return equiLeader;
47     }
48 }
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity: **O(N)**

expand all

Example tests

▶ example	✓ OK
example test	
expand all	Correctness tests
▶ single	✓ OK
single element	
▶ double	✓ OK
two elements	
▶ simple	✓ OK
simple test	
▶ small_random	✓ OK
small random test with two values, length = ~100	
▶ small	✓ OK
random + 200 * [MIN_INT] + random ,length = ~300	
expand all	Performance tests
▶ large_random	✓ OK
large random test with two values, length = ~50,000	
▶ large	✓ OK
random(0,1) + 50000 * [0] + random(0, 1), length = ~100,000	
▶ large_range	✓ OK
1, 2, ..., N, length = ~100,000	
▶ extreme_large	✓ OK
all the same values	