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 ≤ 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:

* 0, because sequences: (4) and (3, 4, 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:

def solution(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].

def solution(A):

dic1 = {}

equileader1 = 0

leader\_cnt1 = 0

for i in A:

if i in dic1:

dic1[i] +=1

if dic1[i] > leader\_cnt1:

leader\_cnt1 = dic1[i]

leader1 = i

else :

dic1[i]=1

if leader\_cnt1<=len(A)/2:

return 0

left\_value\_count = 0

right\_value\_count = leader\_cnt1

for index, value in enumerate(A):

if value == leader1:

left\_value\_count+=1

right\_value\_count-=1

if left\_value\_count>(index+1)/2 and right\_value\_count> (len(A)-index-1)/2:

equileader1 +=1

return equileader1

A = [4, 3, 4, 4, 4, 2]

car1 = solution(A)

print(car1)