Codility_

Screen Report: trainingJPYJBN-EKP

Test Name:

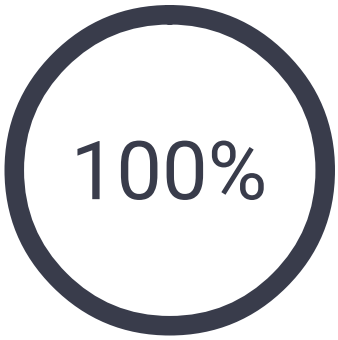
Check out Codility training tasks

Summary Timeline AI Assistant Chat

Tasks summary

Task	Effective time spent	Score
AbsDistinct Python ⚠	19 min	100%

Total score



Assessment integrity

No issues

See less

Identity & network

Signal	Issues 0	Action
Network IP check The candidate used one network during the assessment	Not found	

Tasks Details

1. **AbsDistinct**

Easy

Compute number of distinct absolute values of sorted array elements.

Task Score	Correctness	Performance
100%	100%	100%

Task description

A non-empty array A consisting of N numbers is given. The array is sorted in non-decreasing order. The *absolute distinct count* of this

Solution

Programming language used:	Python
Time spent on task:	19 minutes

array is the number of distinct absolute values among the elements of the array.

For example, consider array A such that:

```
A[0] = -5
A[1] = -3
A[2] = -1
A[3] =  0
A[4] =  3
A[5] =  6
```

The absolute distinct count of this array is 5, because there are 5 distinct absolute values among the elements of this array, namely 0, 1, 3, 5 and 6.

Write a function:

```
def solution(A)
```

that, given a non-empty array A consisting of N numbers, returns absolute distinct count of array A.

For example, given array A such that:

```
A[0] = -5
A[1] = -3
A[2] = -1
A[3] =  0
A[4] =  3
A[5] =  6
```

the function should return 5, 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 [−2,147,483,648..2,147,483,647];
- array A is sorted in non-decreasing order.

Notes: *not defined yet*

Task timeline Beta

Code: 17:00:06 UTC, py, final, score: **100** show code in pop-up

```
1  def solution(A):
2    c = 1                                  #start with co
3    my_max = max(abs(A[0]), abs(A[-1]))    #initialize my
4    index_head = 0                          #Two point, 1
5    index_tail = len(A)-1
6    while index_head <= index_tail:
7      head = abs(A[index_head])            #Take absolute
8      if head == my_max:
9        index_head += 1
10       continue
11     tail = abs(A[index_tail])            #Take absolute
12
13
14     if tail == my_max:
15       index_tail -= 1
16       continue
17     if head >= tail:
18       my_max = head
19       index_head += 1
20     else:
21       my_max = tail
22       index_tail -= 1
23
24     c += 1
25   return c
26
27
28
```

16:41:19 17:00:06

Away from Codility tab 0%

Analysis summary

The solution obtained perfect score.

Analysis

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

expand all	Example tests	
▶ example example test		✓ OK
expand all	Correctness tests	
▶ one_element		✓ OK

▶ two_elements	✓ OK
▶ same_elements	✓ OK
▶ simple	✓ OK
▶ simple_no_zero	✓ OK
▶ simple_no_same	✓ OK
▶ simple_no_negative	✓ OK
▶ simple_no_positive	✓ OK
▶ arith_overlow	✓ OK
▶ medium_chaotic1	✓ OK
▶ medium_chaotic2	✓ OK

expand all Performance tests

▶ long_sequence_no_negative	✓ OK
▶ long_sequence_no_positive	✓ OK
▶ long_sequence	✓ OK