

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

AbsDistinct

START

Compute number of distinct absolute values of sorted array elements.

Programming language: C++ ▼

Human language: English ▼

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

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

that, given a non-empty zero-indexed 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.

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
2015 challenge

Lesson 91

Tasks from
Indeed Prime
2016 challenge

Lesson 92

Tasks from
Indeed Prime
2016 College
Coders
challenge

Lesson 99

Assume that:

- 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.

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