codility

Check out Codility training tasks

Candidate Report: Anonymous

Test Name:

Summary Timeline

Test Score

100 out of 100 points

100%

Tasks in Test

	Time Spent	Task Score
Distinct Submitted in: Scala	3 min	100%

TASKS DETAILS

EASY

1. Distinct

Compute number of distinct values in an array.

Task Score	Correctness	Performance
100%	100%	100%

Task description

Write a function

```
object Solution { def solution(a: Array[Int]): Int }
```

that, given an array A consisting of N integers, returns the number of distinct values in array A.

For example, given array A consisting of six elements such that:

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

the function should return 3, because there are 3 distinct values appearing in array A, namely 1, 2 and 3.

Write an **efficient** algorithm for the following assumptions:

- N is an integer within the range [0..100,000];

Solution

Programming language used:	Scala
Total time used:	3 minutes
Effective time used:	3 minutes
Notes:	*not defined yet*

Task timeline

- each element of array A is an integer within the range [−1,000,000..1,000,000].

07:17:55 07:20:14

Code: 07:20:13 UTC, scala, final, score: **100** show code in pop-up

```scala
import scala.collection.JavaConverters._

// you can write to stdout for debugging purposes, e
// println("this is a debug message")

object Solution {
  def solution(a: Array[Int]): Int = {
    a.toSet.size
  }
}
```

Analysis summary

The solution obtained perfect score.

Analysis

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

expand all	Example tests	
▶	example1 example test, positive answer	✔ OK
expand all	**Correctness tests**	
▶	extreme_empty empty sequence	✔ OK
▶	extreme_single sequence of one element	✔ OK
▶	extreme_two_elems sequence of three distinct elements	✔ OK
▶	extreme_one_value sequence of 10 equal elements	✔ OK
▶	extreme_negative sequence of negative elements, length=5	✔ OK
▶	extreme_big_values sequence with big values, length=5	✔ OK
▶	medium1 chaotic sequence of value sfrom [0..1K], length=100	✔ OK
▶	medium2 chaotic sequence of value sfrom	✔ OK

	[0..1K], length=200	
▶	medium3 chaotic sequence of values from [0..10], length=200	✔ OK

expand all **Performance tests**

▶	large1 chaotic sequence of values from [0..100K], length=10K	✔ OK
▶	large_random1 chaotic sequence of values from [-1M..1M], length=100K	✔ OK
▶	large_random2 another chaotic sequence of values from [-1M..1M], length=100K	✔ OK