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: Java 8	1 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

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

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

Solution

Programming language used:	Java 8
Total time used:	1 minutes
Effective time used:	1 minutes
Notes:	*not defined yet*

Task timeline

18:14:11 18:14:51

Code: 18:14:51 UTC, java, final, score: **100** show code in pop-up

```
1  // you can also use imports, for example:
2  // import java.util.*;
3
4  // you can write to stdout for debugging purposes, e.g.
5  // System.out.println("this is a debug message");
6  import java.util.Hashtable;
7  import java.util.Map;
8
9  class Solution {
```

```
10      public static int solution(int[] a) {
11                  Map<Integer, Integer> m = new Hashtable<In
12
13                  for (int i = 0; i < a.length; i++) {
14                          m.putIfAbsent(a[i], 1);
15                  }
16
17                  int r = m.keySet().size();
18
19                  return r;
20          }
21  }
```

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 [0..1K], length=200	✓ OK
▶	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

PDF version of this report that may be downloaded on top of this site may contain sensitive data including personal information. For security purposes, we recommend you remove it from your system once reviewed.