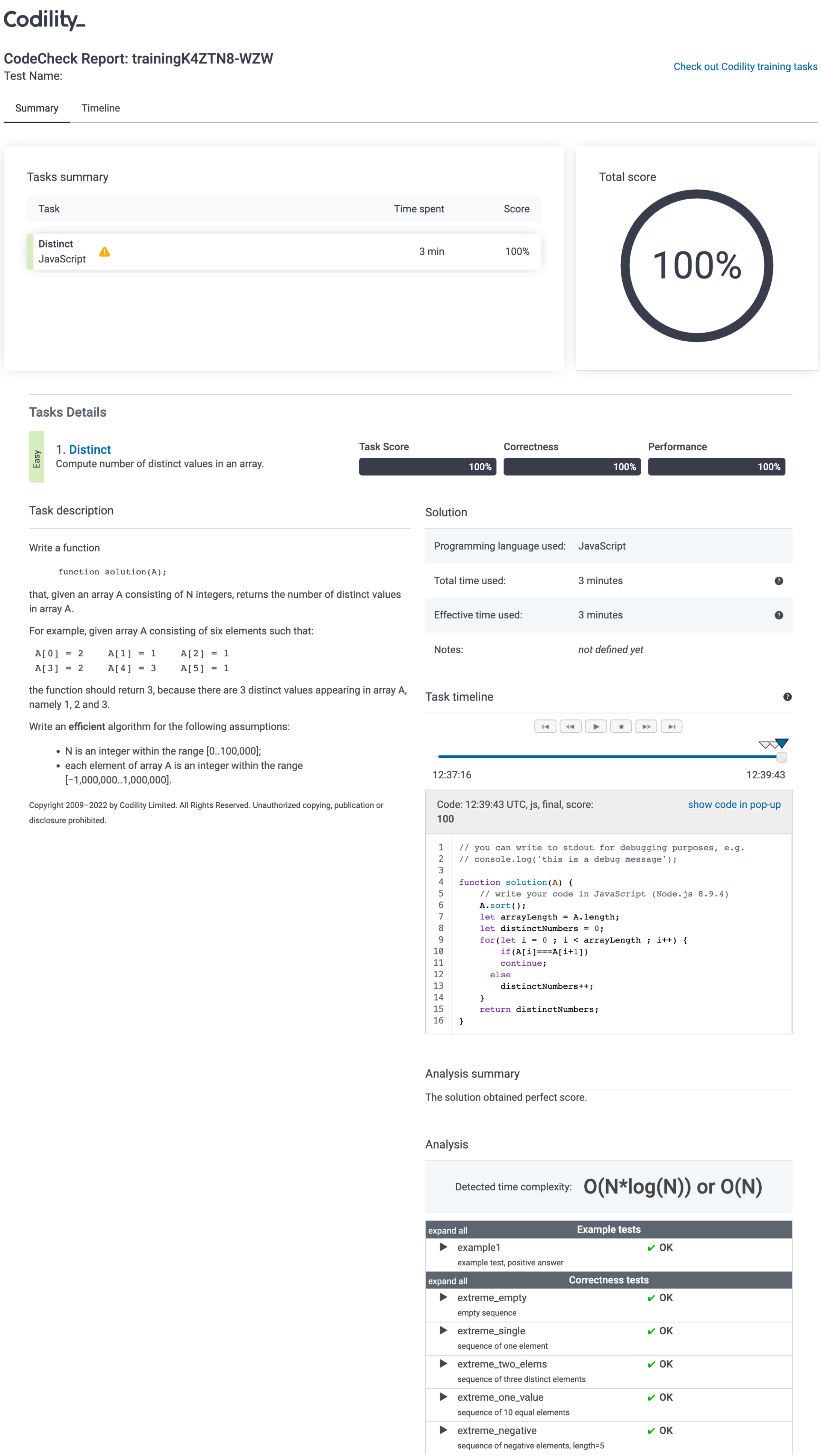


Codility_

CodeCheck Report: trainingK4ZTN8-WZW

Test Name:

Check out Codility training tasks

Summary Timeline

Tasks summary

Task	Time spent	Score
Distinct JavaScript ⚠	3 min	100%

Total score

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

```
function solution(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:	JavaScript
Total time used:	3 minutes
Effective time used:	3 minutes
Notes:	*not defined yet*

Task timeline

12:37:16 12:39:43

Code: 12:39:43 UTC, js, final, score: 100

show code in pop-up

```
// you can write to stdout for debugging purposes, e.g.
// console.log('this is a debug message');

function solution(A) {
    // write your code in JavaScript (Node.js 8.9.4)
    A.sort();
    let arrayLength = A.length;
    let distinctNumbers = 0;
    for(let i = 0 ; i < arrayLength ; i++) {
        if(A[i]===A[i+1])
        continue;
      else
        distinctNumbers++;
    }
    return distinctNumbers;
}
```

Analysis summary

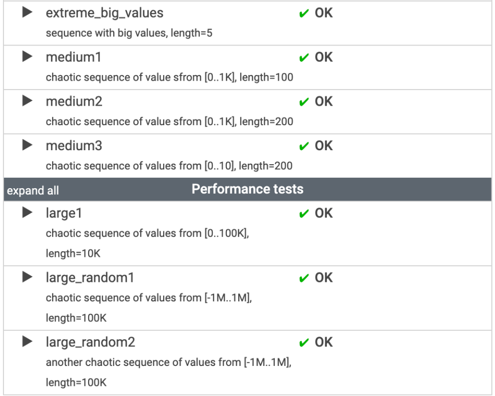
The solution obtained perfect score.

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