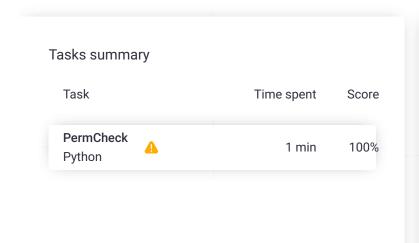
Codility_

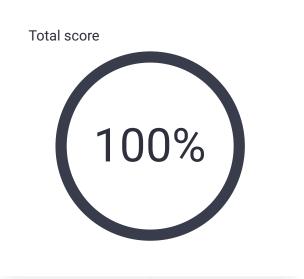
CodeCheck Report: training9NSM5J-4Q8

Test Name:

Check out Codility training tasks

Summary Timeline





Tasks Details

PermCheck
Check Task Score Correctness Performance
whether array 100% 100% 100%
A is a permutation.

Task description

A non-empty array A consisting of N integers is given.

A *permutation* is a sequence containing each element from 1 to N once, and only once.

For example, array A such that:

A[0] = 4

A[1] = 1

A[2] = 3

A[3] = 2

is a permutation, but array A such that:

A[0] = 4

A[1] = 1

Solution

Programming language used: Python

Total time used: 1 minutes

Effective time used: 1 minutes

Notes: not defined yet

Task timeline

1 von 3 17.07.23, 14:06

$$A[2] = 3$$

is not a permutation, because value 2 is missing.

The goal is to check whether array A is a permutation.

Write a function:

that, given an array A, returns 1 if array A is a permutation and 0 if it is not.

For example, given array A such that:

A[0] = 4

A[1] = 1

A[2] = 3

A[3] = 2

the function should return 1.

Given array A such that:

A[0] = 4

A[1] = 1

A[2] = 3

the function should return 0.

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 [1..1,000,000,000].

Copyright 2009–2023 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

```
\nabla
12:05:05
                                      12:05:34
Code: 12:05:33 UTC, py,
                          show code in pop-up
final, score: 100
 1
     # you can write to stdout for debugging
 2
     # print("this is a debug message")
 4
     def solution(A):
 5
         # Implement your solution here
 6
         # pass
7
         length_of_array = len(A)
         sum_of_array = length_of_array * (1
8
         if sum(A) == sum_of_array and len(s
10
11
         return 0
```

Analysis summary

The solution obtained perfect score.

Analysis

 $\begin{array}{c} \text{O(N) or} \\ \text{Detected time complexity:} & \text{O(N *} \\ \text{log(N))} \end{array}$

expand	all	Example tes	ts	
	cample1 e first example t	test	•	ОК
	cample2 e second examp	ole test	•	ОК
expand	all C	Correctness te	est	s
sir	ctreme_min_ ngle element wi nimal/maximal	th	✓	OK
sir	ngle ngle element		•	ОК
,	ouble o elements		•	ОК
tot	ntiSum1 tal sum is corre rmutation, N <=	ct, but it is not a	V	ОК
pe	mall_permuta rmutation + one curs twice, N =	e element	V	ОК
▶ p€	ermutations_	of_ranges	•	ОК

2 von 3 17.07.23, 14:06

permutations of sets like [2100]						
for which the anwsers should be						
false						
expand all Performance t	ests					
medium_permutation permutation + few elements	✓ OK					
occur twice, N = ~10,000						
► antiSum2 total sum is correct, but it is not a permutation, N = ~100,000	✓ OK					
► large_not_permutation permutation + one element occurs three times, N = ~100,000	✓ OK					
► large_range sequence 1, 2,, N, N = ~100,000	✓ OK					
► extreme_values all the same values, N = ~100,000	✓ OK					
various_permutationsall sequences are permutations	✓ OK					

3 von 3