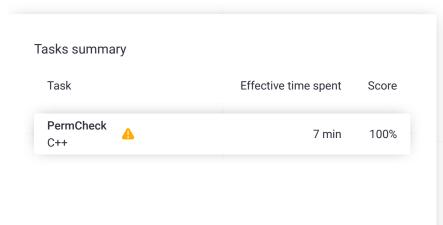
Codility_

Screen Report: Anonymous

Test Name:

Summary Timeline

Check out Codility training tasks





Tasks Details

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

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

A[2] = 3

is not a permutation, because value 2 is missing.

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

Programming language used: C++

Time spent on task: 7 minutes

Notes: not defined yet

Task timeline

12:36:07

Code: 12:42:49 UTC, cpp, final, score: 100

show code in pop-up

1 of 3 9/28/2024, 2:43 PM

Write a function:

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

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 2009–2024 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

```
1
2
3
     #include <vector>
     #include <algorithm>
5
6
     int solution(vector<int> &A) {
7
         // Implement your solution here
8
         int sizeA{(int)A.size()};
9
         vector<int> trackVector(sizeA, 0);
10
11
         for(auto element : A)
12
13
14
             if (element > sizeA || trackVector[element
15
             {
16
                 return 0;
17
18
19
             trackVector[element-1] = 1;
20
         }
21
22
         return 1;
23
     }
24
```

Analysis summary

The solution obtained perfect score.

Detected time complexity:

Analysis

O(N) or O(N * log(N))

expand all	Examp	le tests		
example1 the first exam	ple test	√ OK		
example2 the second ex		√ 0K		
expand all	Correctn	Correctness tests		
extreme_m single elemen value	in_max t with minimal/ma	✓ OK eximal		
single elemen	t	√ OK		
double two elements		√ OK		
antiSum1 total sum is co	orrect, but it is not	√ OK a		

2 of 3 9/28/2024, 2:43 PM

perm	ull_permutation utation + one element occurs s, N = ~100	✓	OK				
•	permutations_of_ranges permutations of sets like [2100] which the anwsers should be fals		√ OK or				
ехра	expand all Performance tests						
>	medium_permutation permutation + few elements occu twice, N = ~10,000	ır	√ OK				
•	antiSum2 total sum is correct, but it is not a permutation, N = ~100,000	ı	√ OK				
>	large_not_permutation permutation + one element occur three times, N = ~100,000	's	√ OK				
•	large_range sequence 1, 2,, N, N = ~100,000)	√ OK				
•	extreme_values all the same values, N = ~100,000)	√ OK				
•	various_permutations all sequences are permutations		√ OK				

3 of 3