Open Feedback Dialog
Open Feedback Dialog

codility

ning center

ut Codility training tasks

Congratulations

You have completed a Codility demo.

Tweet this!
I scored 100% in #ruby on @Codility!
https://codility.com/demo/take-sample-test/perm_check/

Sign up for our newsletter!

Like us on Facebook!

Demo ticket

Session

ID: demoGPYJMS-J3K
Time limit: 120 min.

Status: closed

Created on: 2014-12-14 06:34 UTC
Started on: 2014-12-14 06:34 UTC
Finished on: 2014-12-14 06:51 UTC

Tasks in test	Correctness	Performance	Task score
1 PermCheck	100%	100%	100%

Test score

100%

100 out of 100 points

EASY

1. PermCheck

Check whether array A is a permutation.

score: 100 of 100

Task description

A non-empty zero-indexed 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.
Write a function:

```
def solution(a)
```

that, given a zero-indexed 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
```

Solution

Programming language used: Ruby

Total time used: 18 minutes

Effective time used: 18 minutes

Notes: *not defined yet*

Task timeline

06:34:40 06:51:50

Code: 06:51:50 UTC, rb, final, score: **100.00**

```
# you can use puts for debugging purposes, e.g.
# puts "this is a debug message"

def solution(a)
  h = Hash.new(0)
  a.each do |n|
    h[n] += 1
    # not a permutation if more than one occurrrence
    return 0 if h[n] != 1
  end
  (1..a.size).each do |n|
    # not a permutation if no occurrences
    return 0 if h[n] != 1
  end
  1
```

the function should return 0.
Assume that:

- 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].

Complexity:

- expected worst-case time complexity is O(N);
- expected worst-case space complexity is O(N), beyond input storage (not counting the storage required for input arguments).

Elements of input arrays can be modified.

```
end
```

Analysis

Detected time complexity:

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

test	time	result
Example tests		
example1 the first example test	0.068 s	**OK**
example2 the second example test	0.068 s	**OK**
Correctness tests		
extreme_min_max single element with minimal/maximal value	0.052 s	**OK**
single single element	0.068 s	**OK**
double two elements	0.056 s	**OK**
antiSum1 total sum is correct, but it is not a permutation, N <= 10	0.056 s	**OK**
small_permutation permutation + one element occurs twice, N = ~100	0.056 s	**OK**
Performance tests		
medium_permutation permutation + few elements occur twice, N = ~10,000	0.072 s	**OK**
antiSum2 total sum is correct, but it is not a permutation, N = ~100,000	0.264 s	**OK**
large_permutation permutation + one element occurs three times, N = ~100,000	0.240 s	**OK**
large_range sequence 1, 2, ..., N, N = ~100,000	0.264 s	**OK**
extreme_values all the same values, N = ~100,000	0.052 s	**OK**

Training center