codility

Training center

Check out Codility training tasks

Congratulations

You have completed a Codility demo.

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

Sign up for our newsletter!

Like us on Facebook!

Demo ticket

Session

ID: demoPBGNPV-KR5
Time limit: 120 min.

Status: closed

Created on: 2014-04-24 17:38 UTC
Started on: 2014-04-24 17:38 UTC
Finished on: 2014-04-24 18:22 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.
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:

Solution

Programming language used: Python

Total time used: 44 minutes

Effective time used: 44 minutes

Notes: correct functionality and scalability

Task timeline

17:38:51 — 18:22:45

Code: 18:22:45 UTC, py, final, score: **100.00**

```
01. def solution(A):
02.     n = len(A)
03.     count = [0] * (n+1)
04.     for v in A:
05.         # out of range
06.         if v > n:
07.             return 0
08.         # duplicate
09.         elif count[v]:
10.             return 0
11.         else:
12.             count[v] += 1
13.
14.     return 1
```

```
A[0] = 4
A[1] = 1
A[2] = 3
```

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.

Analysis

Detected time complexity:
O(N) or O(N * log(N))

test	time	result
Example tests		
example1 the first example test	0.050 s.	**OK**
example2 the second example test	0.050 s.	**OK**
Correctness tests		
extreme_max single element with maximal value	0.050 s.	**OK**
single single element	0.050 s.	**OK**
double two elements	0.050 s.	**OK**
antiSum1 total sum is corret (equals 1 + 2 + ... N), but it is not a permutation, N = 3	0.050 s.	**OK**
Performance tests		
medium_permutation permutation, N = ~10,000	0.060 s.	**OK**
antiSum2 total sum is corret (equals 1 + 2 + ... N), but it is not a permutation, N = ~100,000	0.230 s.	**OK**
large_permutation large permutation, N = ~100,000	0.230 s.	**OK**
large_range sequence 1, 2, ..., N, N = ~100,000	0.240 s.	**OK**
extreme_values all the same values, N = ~100,000	0.200 s.	**OK**

Training center