Open Feedback Dialog

c0d1l1ty
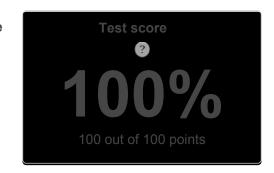
Demo ticket

Session

ID: demoG7PKE5-X4Q
Time limit: 120 min.

Status: closed

Created on: 2014-03-17 07:52 UTC
Started on: 2014-03-17 07:54 UTC
Finished on: 2014-03-17 07:57 UTC

Tasks in test

Task score

Test score

100%

100 out of 100 points

EASY

1. PrefixSet

Given a table A of N integers from 0 to N-1 calculate the smallest such index P, that that {A[0],...,A[N-1]} = {A[0],...,A[P]}.

score: 100 of 100

Task description

A non-empty zero-indexed array A consisting of N integers is given. The *first covering prefix* of array A is the smallest integer P such that $0 \le P < N$ and such that every value that occurs in array A also occurs in sequence A[0], A[1], ..., A[P].
For example, the first covering prefix of the following 5−element array A:

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

is 3, because sequence [A[0], A[1], A[2], A[3]] equal to [2, 2, 1, 0], contains all values that occur in array A.

Write a function

```
def solution(A)
```

that, given a zero-indexed non-empty array A consisting of N integers, returns the first covering prefix of A.
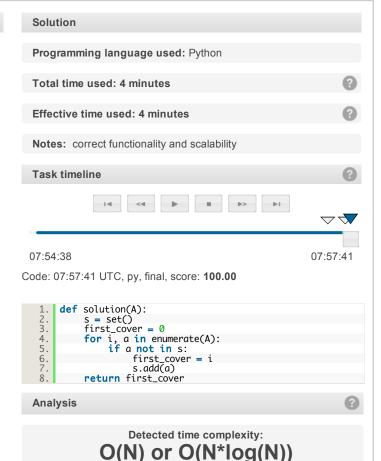For example, given array A such that

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

the function should return 3, as explained above.
Assume that:

- N is an integer within the range [1..1,000,000];
- each element of array A is an integer within the range [0..N−1].

Complexity:

- expected worst-case time complexity is O(N);

Solution

Programming language used: Python

Total time used: 4 minutes

Effective time used: 4 minutes

Notes: correct functionality and scalability

Task timeline

07:54:38   07:57:41

Code: 07:57:41 UTC, py, final, score: **100.00**

```python
1.  def solution(A):
2.      s = set()
3.      first_cover = 0
4.      for i, a in enumerate(A):
5.          if a not in s:
6.              first_cover = i
7.          s.add(a)
8.      return first_cover
```

Analysis

Detected time complexity:

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

test	time	result
example1 example test	0.050 s.	OK

- 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.

extreme_single 1-element sequence	0.050 s.	OK
extreme_two 1-element sequence	0.050 s.	OK
extreme_constant constant sequence	0.050 s.	OK
extreme_identity identity permutation	0.050 s.	OK
extreme_permutation permutation	0.050 s.	OK
simple1 very simple sequence	0.050 s.	OK
simple2 simple sequence	0.050 s.	OK
binary binary sequence	0.050 s.	OK
periodic periodic pattern	0.050 s.	OK
random_dec_100 random test 100 elements, 37 different values.	0.050 s.	OK
random_dec_1000 random test 1000 elements, 34 different values.	0.050 s.	OK
random_dec_10000 random test 10 000 elements, 30 different values.	0.060 s.	OK
random_dec_100000 random test 100 000 elements, 27 different values.	0.210 s.	OK
random_sqrt_100 random test 100 elements, 10 different values.	0.050 s.	OK
random_sqrt_1000 random test 1000 elements, 31 different values.	0.050 s.	OK
random_sqrt_10000 random test 10 000 elements, 100 different values.	0.070 s.	OK
random_sqrt_100000 random test 100 000 elements, 316 different values.	0.230 s.	OK
random_n_log_100 random test 100 elements and n/log_2 n values.	0.050 s.	OK
random_n_log_1000 random test 1000 elements and n/log_2 n values.	0.050 s.	OK
random_n_log_10000 random test 10 000 elements and n/log_2 n values.	0.060 s.	OK
random_n_log_100000 random test 100 000 elements and n/log_2 n values.	0.230 s.	OK
random_n_100 random test 100 elements and values.	0.050 s.	OK
random_n_1000 random test 1000 elements and values.	0.050 s.	OK
random_n_10000 random test 10 000 elements and values.	0.070 s.	OK
random_n_100000 random test 100 000 elements and values.	0.240 s.	OK

Training center