Open Feedback Dialog

codility

Training center

Check out Codility training tasks

Demo ticket

Session

ID: demoEBRF39-QZF
Time limit: 120 min.

Status: closed

Created on: 2014-03-17 18:44 UTC
Started on: 2014-03-17 18:45 UTC
Finished on: 2014-03-17 18:45 UTC

Tasks in test

Task score

Test score

?

100%

100 out of 100 points

MEDIUM

1. MinAvgTwoSlice

Find the minimal average of any slice containing at least two elements.

score: 100 of 100

Task description

A non-empty zero-indexed array A consisting of N integers is given. A pair of integers (P, Q), such that 0 ≤ P < Q < N, is called a *slice* of array A (notice that the slice contains at least two elements). The *average* of a slice (P, Q) is the sum of A[P] + A[P + 1] + ... + A[Q] divided by the length of the slice. To be precise, the average equals (A[P] + A[P + 1] + ... + A[Q]) / (Q − P + 1).
For example, array A such that:

```
A[0] = 4
A[1] = 2
A[2] = 2
A[3] = 5
A[4] = 1
A[5] = 5
A[6] = 8
```

contains the following example slices:

- slice (1, 2), whose average is (2 + 2) / 2 = 2;
- slice (3, 4), whose average is (5 + 1) / 2 = 3;
- slice (1, 4), whose average is (2 + 2 + 5 + 1) / 4 = 2.5.

The goal is to find the starting position of a slice whose average is minimal.
Write a function:

```
def solution(A)
```

that, given a non-empty zero-indexed array A consisting of N integers, returns the starting position of the slice with the minimal average. If there is more than one slice with a minimal average, you should return the smallest starting position of such a slice.
For example, given array A such that:

```
A[0] = 4
A[1] = 2
A[2] = 2
A[3] = 5
A[4] = 1
A[5] = 5
```

Solution

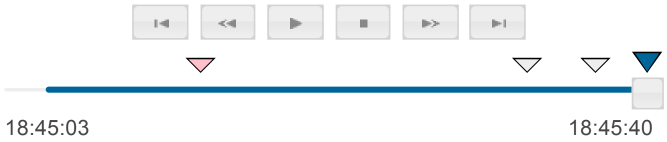
Programming language used: Python

Total time used: 1 minutes ?

Effective time used: 1 minutes ?

Notes: correct functionality and scalability

Task timeline ?



Code: 18:45:40 UTC, py, final, score: **100.00**

```
def solution(A):
    N = len(A)
    avg = [0] * (N-1)
    last = N - 1
    for i in xrange(N-2, -1, -1):
        avg[i] = (A[i] + A[i+1]) / 2.0
        if i < N-2:
            x =  1.0 * (A[i] + avg[i+1] * (last - i)) / (last - i + 1)
            if x < avg[i]:
                avg[i] = x
            else:
                last = i + 1

    min_avg = sys.maxint
    min_index = -1
    for i in xrange(N-1):
        if avg[i] < min_avg:
            min_avg = avg[i]
            min_index = i
    return min_index
    A[6] = 8
```

the function should return 1, as explained above.
Assume that:

- N is an integer within the range [2..100,000];
- each element of array A is an integer within the range [−10,000..10,000].

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)

test	time	result
example example test	0.050 s.	**OK**
double_quadruple two or four elements	0.050 s.	**OK**
simple1 simple test, the best slice has length 3	0.050 s.	**OK**
simple2 simple test, the best slice has length 3	0.050 s.	**OK**
small_random random, length = 100	0.050 s.	**OK**
medium_range increasing, decreasing (legth = ~100) and small functional	0.050 s.	**OK**
medium_random random, N = ~700	0.050 s.	**OK**
large_ones numbers from -1 to 1, N = ~100,000	0.280 s.	**OK**
large_random random, N = ~100,000	0.350 s.	**OK**
extreme_values all maximal values, N = ~100,000	0.340 s.	**OK**
large_sequence many seqeneces, N = ~100,000	0.250 s.	**OK**

Training center