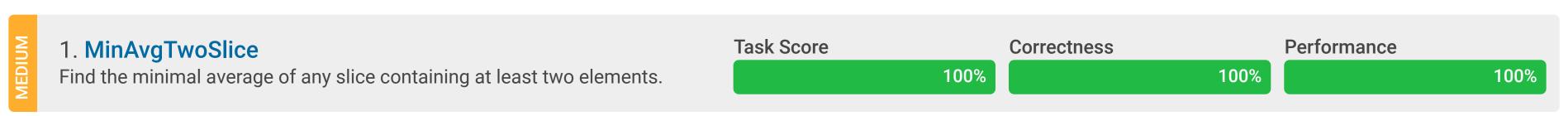
Check out Codility training tasks

TASKS DETAILS



Task description

A non-empty zero-indexed array A consisting of N integers is given. A pair of integers (P, Q), such that $0 \le P < Q < N$, 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] = 4A[1] = 2A[2] = 2A[3] = 5A[4] = 1A[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:

class Solution { public int solution(int[] 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] = 4A[1] = 2A[2] = 2A[3] = 5A[4] = 1A[5] = 5A[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).

Copyright 2009–2018 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Solution



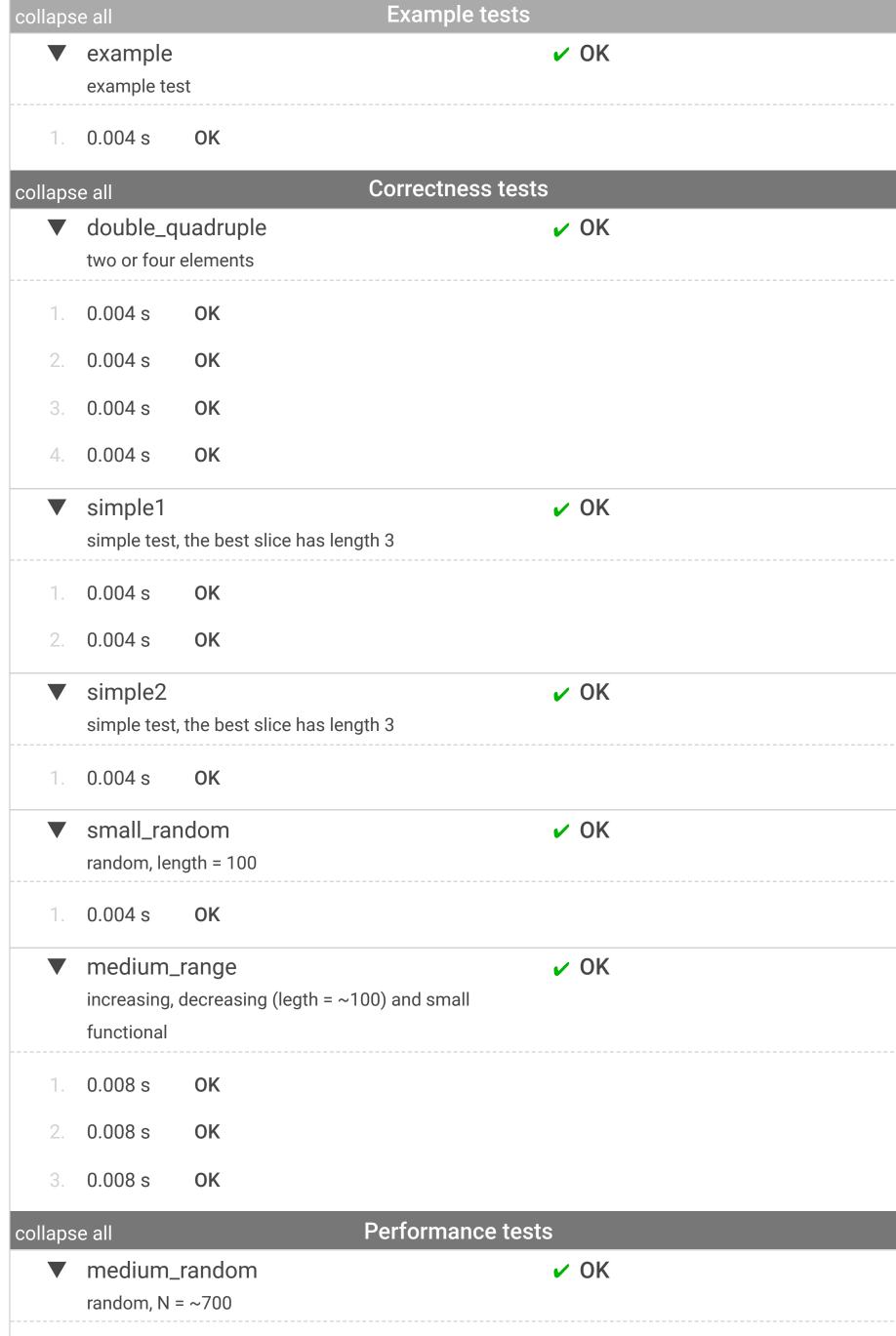
Analysis summary

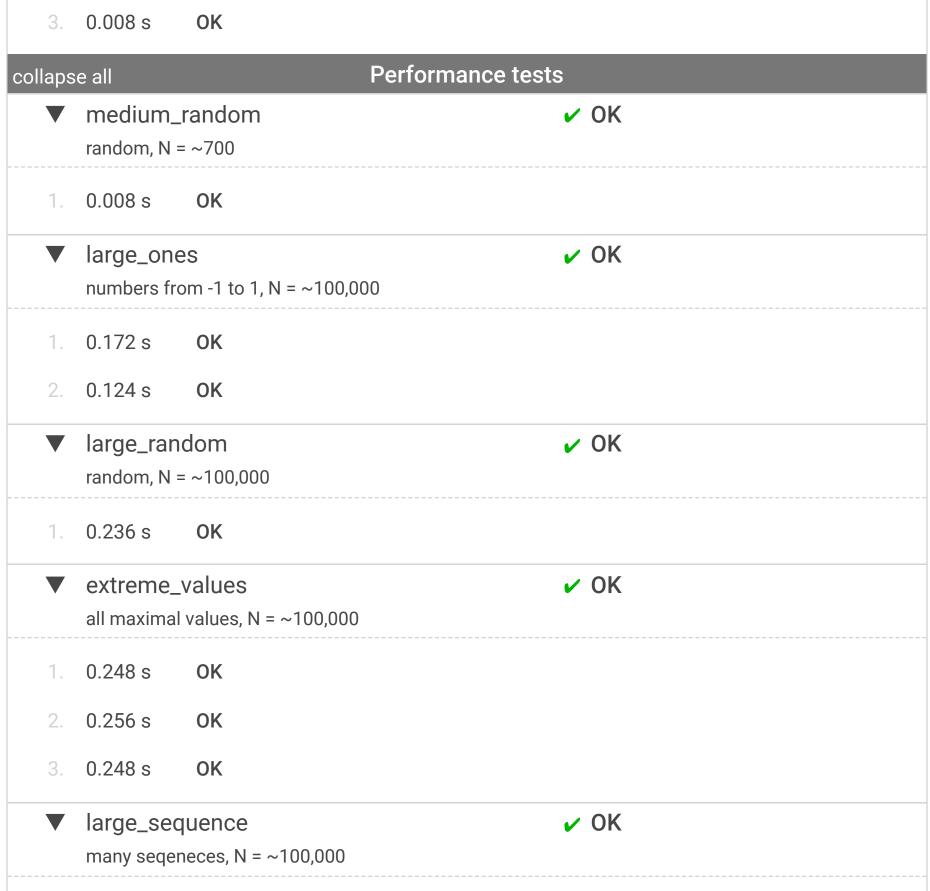
The solution obtained perfect score.

Analysis ?

45







1. 0.176 s

2. 0.128 s

OK

OK