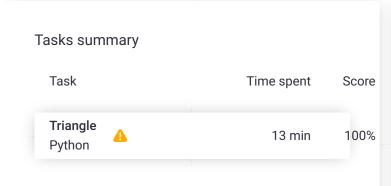
Codility_

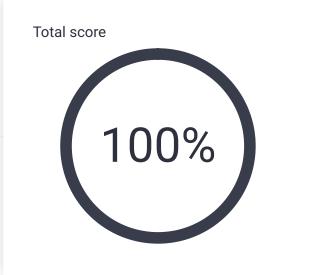
CodeCheck Report: training225D8P-DYU

Test Name:

Check out Codility training tasks

Summary Timeline





Tasks Details

1.

Triangle

Determine whether a triangle can be

built from a given set of edges. Task Score

Correctness

100%

Performance

100%

100%

Task description

An array A consisting of N integers is given. A triplet (P, Q, R) is triangular if $0 \le P < Q < R < N$ and:

- A[P] + A[Q] > A[R],
- A[Q] + A[R] > A[P],
- A[R] + A[P] > A[Q].

For example, consider array A such that:

Solution

Programming language used: Python

Total time used: 13 minutes **3**

Effective time used: 13 minutes 3

Notes: not defined yet

1 von 3

$$A[0] = 10$$
 $A[1] = 2$ $A[2] = 5$
 $A[3] = 1$ $A[4] = 8$ $A[5] = 20$

Triplet (0, 2, 4) is triangular.

Write a function:

that, given an array A consisting of N integers, returns 1 if there exists a triangular triplet for this array and returns 0 otherwise.

For example, given array A such that:

$$A[0] = 10$$
 $A[1] = 2$ $A[2] = 5$
 $A[3] = 1$ $A[4] = 8$ $A[5] = 20$

the function should return 1, as explained above. Given array A such that:

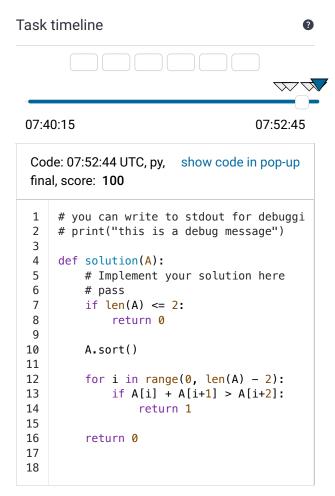
$$A[0] = 10$$
 $A[1] = 50$ $A[2] = 5$ $A[3] = 1$

the function should return 0.

Write an **efficient** algorithm for the following assumptions:

- N is an integer within the range [0..100,000];
- each element of array A is an integer within the range [-2,147,483,648..2,147,483,647].

Copyright 2009–2023 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.



Analysis summary

The solution obtained perfect score.

Analysis

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

expand all Example	tests
example	✓
example, positive answer,	ОК
length=6	
► example1	✓
example, answer is zero,	OK
length=4	
expand all Correctnes	s tests
extreme_empty	✓
empty sequence	ОК
extreme_single	✓
1-element sequence	ОК

2 von 3 18.07.23, 09:54

2-el	ement sequence OK	(
•	extreme_negative1	✓
	three equal negative numbers	ок
•	extreme_arith_overflow1	V
	overflow test, 3 MAXINTs	ОК
•	extreme_arith_overflow2	<i>V</i>
	overflow test, 10 and 2 MININTs	, OK
	extreme_arith_overflow3	✓
	overflow test, 0 and 2 MAXINTs	OK
	medium1	V
	chaotic sequence of values	OK
	from [0100K], length=30	
	medium2	✓
	chaotic sequence of values	ОК
	from [01K], length=50	
>	medium3	✓
	chaotic sequence of values	ОК
	from [01K], length=100	
ехр	and all Performance	tests
exp	and all Performance large1	tests 🗸
exp		tests V OK
exp	large1	✓
exp •	large1 chaotic sequence with values	✓
exp •	large1 chaotic sequence with values from [0100K], length=10K	ОК
exp	large1 chaotic sequence with values from [0100K], length=10K large2	ok
e xp	large1 chaotic sequence with values from [0100K], length=10K large2 1 followed by an ascending	ok
exp ▶	large1 chaotic sequence with values from [0100K], length=10K large2 1 followed by an ascending sequence of ~50K elements	ok
exp	large1 chaotic sequence with values from [0100K], length=10K large2 1 followed by an ascending sequence of ~50K elements from [0100K], length=~50K	ok
e xp	large1 chaotic sequence with values from [0100K], length=10K large2 1 followed by an ascending sequence of ~50K elements from [0100K], length=~50K large_random	OK OK
exp ▶ ▶	large1 chaotic sequence with values from [0100K], length=10K large2 1 followed by an ascending sequence of ~50K elements from [0100K], length=~50K large_random chaotic sequence of values	OK OK
exp ▶	large1 chaotic sequence with values from [0100K], length=10K large2 1 followed by an ascending sequence of ~50K elements from [0100K], length=~50K large_random chaotic sequence of values from [01M], length=100K	OK OK
exp ▶ ▶	large1 chaotic sequence with values from [0100K], length=10K large2 1 followed by an ascending sequence of ~50K elements from [0100K], length=~50K large_random chaotic sequence of values from [01M], length=100K large_negative	OK OK OK
exp	large1 chaotic sequence with values from [0100K], length=10K large2 1 followed by an ascending sequence of ~50K elements from [0100K], length=~50K large_random chaotic sequence of values from [01M], length=100K large_negative chaotic sequence of negative	OK OK OK
exp ▶ ▶	large1 chaotic sequence with values from [0100K], length=10K large2 1 followed by an ascending sequence of ~50K elements from [0100K], length=~50K large_random chaotic sequence of values from [01M], length=100K large_negative chaotic sequence of negative values from [-1M1],	OK OK OK
exp ▶ ▶	large1 chaotic sequence with values from [0100K], length=10K large2 1 followed by an ascending sequence of ~50K elements from [0100K], length=~50K large_random chaotic sequence of values from [01M], length=100K large_negative chaotic sequence of negative values from [-1M1], length=100K	OK OK OK
 exp ▶ 	large1 chaotic sequence with values from [0100K], length=10K large2 1 followed by an ascending sequence of ~50K elements from [0100K], length=~50K large_random chaotic sequence of values from [01M], length=100K large_negative chaotic sequence of negative values from [-1M1], length=100K large_negative2	OK OK OK OK
exp	large1 chaotic sequence with values from [0100K], length=10K large2 1 followed by an ascending sequence of ~50K elements from [0100K], length=~50K large_random chaotic sequence of values from [01M], length=100K large_negative chaotic sequence of negative values from [-1M1], length=100K large_negative2 chaotic sequence of negative	OK OK OK OK
exp ►	large1 chaotic sequence with values from [0100K], length=10K large2 1 followed by an ascending sequence of ~50K elements from [0100K], length=~50K large_random chaotic sequence of values from [01M], length=100K large_negative chaotic sequence of negative values from [-1M1], length=100K large_negative2 chaotic sequence of negative values from [-101],	OK OK OK OK
 exp ▶ 	large1 chaotic sequence with values from [0100K], length=10K large2 1 followed by an ascending sequence of ~50K elements from [0100K], length=~50K large_random chaotic sequence of values from [01M], length=100K large_negative chaotic sequence of negative values from [-1M1], length=100K large_negative2 chaotic sequence of negative values from [-101], length=100K	OK OK OK OK

3 von 3