cødılıty

Candidate Report: Anonymous

Test Name:

Summary   Timeline

Test Score

0 out of 100 points

0%

Tasks in Test

	Time Spent ⓘ	Task Score
Triangle Submitted in: Java 8	30 min	0%

TASKS DETAILS

EASY

1. Triangle
Determine whether a triangle can be built from a given set of edges.

Task Score	Correctness	Performance ❓
0%	0%	0%

Task description

An array A consisting of N integers is given. A triplet (P, Q, R) is *triangular* if 0 ≤ 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:

```
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:

```
class Solution { public int solution(int[] A); }
```

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
```
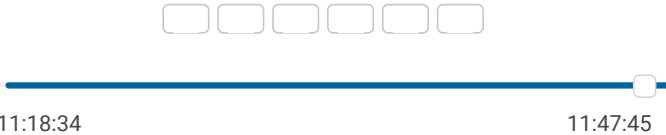
the function should return 0.

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

- N is an integer within the range [0..100,000];

Solution

Programming language used:	Java 8
Total time used:	30 minutes
Effective time used:	30 minutes
Notes:	*not defined yet*

Task timeline ❓

☐ ☐ ☐ ☐ ☐ ☐

11:18:34   11:47:45

Code: 11:47:44 UTC, java, final, score: **0**

show code in pop-up

```
1    // you can also use imports, for example:
2    // import java.util.*;
3
4    // you can write to stdout for debugging purposes, e.g.
5    // System.out.println("this is a debug message");
6
7    class Solution {
8        public static int solution(int[] a) {
9            if (a.length < 3)
```

- each element of array A is an integer within the range [−2,147,483,648..2,147,483,647].

```
10                        return 0;
11              for (int p = 0; p < a.length - 3; p++) {
12                      //System.out.println(p);
13
14                      for (int q = p + 1; q < a.length;
15                              //System.out.println(p + "
16
17                              for (int r = q + 1; r < a.
18                                      //System.out.print
19                                      if (ck(a[p], a[q],
20                                              return 1;
21                              }
22                      }
23              }
24              return 0;
25          }
26
27          public static boolean ck(int ap, int aq, int ar) {
28              return ap + aq > ar && aq + ar > ap && ar
29          }
30      }
```

Analysis summary

The following issues have been detected: wrong answers, timeout errors.

For example, for the input [5, 3, 3] the solution returned a wrong answer (got 0 expected 1).

Analysis ❓

Example tests		
expand all		
▶ example	✓ OK	
example, positive answer, length=6		
▶ example1	✓ OK	
example, answer is zero, length=4		

Correctness tests		
expand all		
▶ extreme_empty	✗ WRONG ANSWER	
empty sequence	got 0 expected 1	
▶ extreme_single	✗ WRONG ANSWER	
1-element sequence	got 0 expected 1	
▶ extreme_two_elems	✗ WRONG ANSWER	
2-element sequence	got 0 expected 1	
▶ extreme_negative1	✗ WRONG ANSWER	
three equal negative numbers	got 0 expected 1	
▶ extreme_arith_overflow1	✗ WRONG ANSWER	
overflow test, 3 MAXINTs	got 0 expected 1	
▶ extreme_arith_overflow2	✗ WRONG ANSWER	
overflow test, 10 and 2 MININTs	got 0 expected 1	
▶ extreme_arith_overflow3	✗ WRONG ANSWER	
overflow test, 0 and 2 MAXINTs	got 0 expected 1	
▶ medium1	✗ WRONG ANSWER	
chaotic sequence of values from [0..100K], length=30	got 0 expected 1	
▶ medium2	✗ WRONG ANSWER	
chaotic sequence of values from [0..1K], length=50	got 0 expected 1	
▶ medium3	✗ WRONG ANSWER	
chaotic sequence of values from [0..1K], length=100	got 0 expected 1	

Performance tests		
expand all		
▶ large1	✗ WRONG ANSWER	
chaotic sequence with values from	got 0 expected 1	

[0..100K], length=10K

▶	large2		✗	**TIMEOUT ERROR**
	1 followed by an ascending sequence of ~50K elements from [0..100K], length=~50K			Killed. Hard limit reached: 6.000 sec.
▶	large_random		✗	**WRONG ANSWER**
	chaotic sequence of values from [0..1M], length=100K			got 0 expected 1
▶	large_negative		✗	**TIMEOUT ERROR**
	chaotic sequence of negative values from [-1M..-1], length=100K			Killed. Hard limit reached: 7.000 sec.
▶	large_negative2		✗	**TIMEOUT ERROR**
	chaotic sequence of negative values from [-10..-1], length=100K			Killed. Hard limit reached: 6.000 sec.
▶	large_negative3		✗	**TIMEOUT ERROR**
	sequence of -1 value, length=100K			Killed. Hard limit reached: 6.000 sec.

PDF version of this report that may be downloaded on top of this site may contain sensitive data including personal information. For security purposes, we recommend you remove it from your system once reviewed.