



Congratulations

You have completed a Codility training test.

Tweet this!

I scored 93% in #cpp on @Codility!
<https://codility.com/demo/take-sample-test/triangle/>

Sign up for our newsletter!

Like us on Facebook!

Training ticket

Session

ID: trainingZB9DZT-YCY
Time limit: 120 min.

Status: closed

Created on: 2018-01-21 06:21 UTC
Started on: 2018-01-21 06:21 UTC
Finished on: 2018-01-21 06:24 UTC

Tasks in test

1 | Triangle
Submitted in: C++

Correctness

90%

Performance

100%

Task score

93%

Test score

93%

93 out of 100 points

EASY

1. Triangle

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

score: 93 of 100

Task description

A zero-indexed array A consisting of N integers is given. A triplet (P, Q, R) is *triangular* if $0 \leq P < Q < R < N$ 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:

```
int solution(vector<int> &A);
```

that, given a zero-indexed 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.

Assume that:

- 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].

Solution

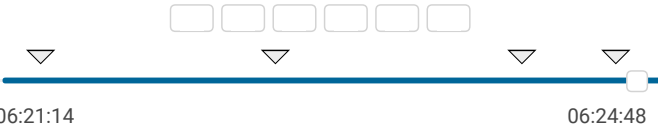
Programming language used: C++

Total time used: 4 minutes

Effective time used: 4 minutes

Notes: *not defined yet*

Task timeline



Code: 06:24:47 UTC, cpp, final,
score: 93

[show code in pop-up](#)

```
1 // you can use includes, for example:
2 #include <algorithm>
3
4 // you can write to stdout for debugging purposes, e. g.
5 // cout << "this is a debug message" << endl;
6
7 int solution(vector<int> &A) {
8     // write your code in C++14 (g++ 6.2.0)
9     int size = A.size();
10    int ret = 0;
11    int minPositive = 0;
12    int positiveStart = 0;
13    int positiveNums = 0;
14    int i = 0;
15    int leftSide = 0;
16    if (size < 3)
```

Complexity:

- expected worst-case time complexity is $O(N \cdot \log(N))$;
- 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.

```
17     {
18         ret = 0;
19         goto exit;
20     }
21
22     sort(&A[0], &A[0] + size);
23     //std::qsort(&A[0], size, sizeof(int));
24     i = 0;
25     while (A[i] < 0)
26     {
27         i++;
28     }
29     positiveStart = i;
30     positiveNums = size - i;
31
32     if (positiveStart >= size-2)
33     {
34         ret = 0;
35         goto exit;
36     }
37
38
39     for (leftSide = positiveStart; leftSide < size - 2; leftSide
40     {
41         long long temp = A[leftSide] + A[leftSide + 1];
42         if (temp > A[leftSide + 2])
43             break;
44     }
45     if (leftSide == size - 2)
46         ret = 0;
47     else
48         ret = 1;
49 exit:
50     return ret;
51
52 }
```

Analysis summary

The following issues have been detected: wrong answers.

Analysis



Detected time complexity:
 $O(N \cdot \log(N))$

Example tests	
▶ example	✓ OK
example, positive answer, length=6	
▶ example1	✓ OK
example, answer is zero, length=4	
Correctness tests	
▶ extreme_empty	✓ OK
empty sequence	
▶ extreme_single	✓ OK
1-element sequence	
▶ extreme_two_elems	✓ OK
2-element sequence	
▶ extreme_negative1	✓ OK
three equal negative numbers	
▶ extreme_arith_overflow1	✗ WRONG ANSWER
overflow test, 3 MAXINTs	
got 0 expected 1	
▶ extreme_arith_overflow2	✓ OK
overflow test, 10 and 2 MININTs	
▶ extreme_arith_overflow3	✓ OK
overflow test, 0 and 2 MAXINTs	
▶ medium1	✓ OK
chaotic sequence of values from [0..100K], length=30	
▶ medium2	✓ OK
chaotic sequence of values from [0..1K], length=50	
▶ medium3	✓ OK
chaotic sequence of values from [0..1K],	

length=100		
expand all		
Performance tests		
▶ large1	chaotic sequence with values from [0..100K], length=10K	✓ OK
▶ large2	1 followed by an ascending sequence of ~50K elements from [0..100K], length=~50K	✓ OK
▶ large_random	chaotic sequence of values from [0..1M], length=100K	✓ OK
▶ large_negative	chaotic sequence of negative values from [-1M..-1], length=100K	✓ OK
▶ large_negative2	chaotic sequence of negative values from [-10..-1], length=100K	✓ OK
▶ large_negative3	sequence of -1 value, length=100K	✓ OK

Training center

How likely are you to recommend Codility to your friends and colleagues?



Not at all likely

Extremely likely