## Codility\_

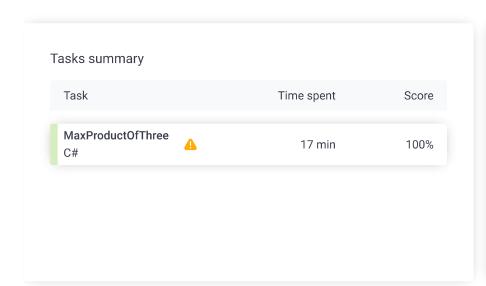
### CodeCheck Report: trainingN447RN-TVC

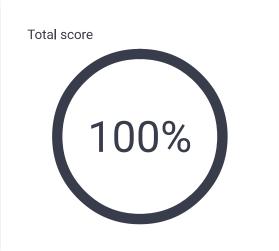
Test Name:

Summary

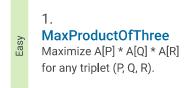
Timeline

Check out Codility training tasks





#### **Tasks Details**



Task Score

Correctness 100%

Performance

100%

#### Task description

A non-empty array A consisting of N integers is given. The *product* of triplet (P, Q, R) equates to A[P] \* A[Q] \* A[R] ( $0 \le P < Q < R < N$ ).

For example, array A such that:

A[0] = -3

A[1] = 1

A[2] = 2

A[3] = -2

A[4] = 5

A[5] = 6

contains the following example triplets:

- (0, 1, 2), product is -3 \* 1 \* 2 = -6
- (1, 2, 4), product is 1 \* 2 \* 5 = 10
- (2, 4, 5), product is 2 \* 5 \* 6 = 60

Your goal is to find the maximal product of any triplet.

Write a function:

#### Solution

olution			
Programming language used: C#			
Total time used: 17 minutes	•		
Effective time used: 17 minutes	•		
Notes: not defined yet			
Task timeline ②			
17:42:34			
	17:59:25		

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

that, given a non-empty array A, returns the value of the maximal product of any triplet.

For example, given array A such that:

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

the function should return 60, as the product of triplet (2, 4, 5) is maximal

Write an efficient algorithm for the following assumptions:

- N is an integer within the range [3..100,000];
- each element of array A is an integer within the range [-1,000..1,000].

Copyright 2009–2022 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

```
Code: 17:59:24 UTC, cs, final,
                                      show code in pop-up
score: 100
1
     using System;
2
3
      * 6.2 - Max Product of Three
4
5
      * Paulo Santos
 6
      * 07.Dec.2022
 7
8
     class Solution {
         public int solution(int[] A) {
9
10
11
              * Check the input
12
              */
13
14
              if (A == null)
15
                  throw new ArgumentNullException();
16
              if (A.Length < 3)
17
                  throw new ArgumentException();
18
              var min1 = int.MaxValue;
19
20
              var min2 = int.MaxValue;
21
              var max1 = int.MinValue;
              var max2 = int.MinValue;
22
23
              var max3 = int.MinValue;
24
25
              for (var i = 0; i < A.Length; i++) {
26
                   * Get the two smallest numbers
27
28
                   * in the array.
29
                   */
                  if (A[i] < min1) {</pre>
30
                      min2 = min1;
31
32
                      min1 = A[i];
33
34
                  else if (A[i] < min2) {</pre>
                      min2 = A[i];
35
36
                  }
37
                  /*
                   * Get the three largest numbers
38
                   * in the array.
39
40
                   */
41
                  if (A[i] > max1) {
42
                      max3 = max2;
43
                      max2 = max1;
                      max1 = A[i];
44
45
                  } else if (A[i] > max2) {
46
                      max3 = max2;
47
                      max2 = A[i];
                  } else if (A[i] > max3) {
48
49
                      max3 = A[i];
50
                  }
51
              }
52
53
              return Math.Max(min1 * min2 * max1, max1 *
54
```

#### Analysis summary

}

The solution obtained perfect score.

#### Analysis

55

56 }

Detected time complexity:

# O(N \* log(N))

ехра	nd all Ex	kample tests	
•	example example test	√ OK	
expand all Correctness tests			
•	one_triple three elements	✓ OK	
•	simple1 simple tests	√ OK	
•	simple2 simple tests	✓ OK	
•	small_random random small, length = 10	<b>✓ OK</b>	
expand all Performance tests			
•	medium_range -1000, -999, 1000, lengt	<b>✓ OK</b> h = ~1,000	
•	medium_random random medium, length =	✓ <b>OK</b> ~10,000	
•	large_random random large, length = ~1	<b>✓ OK</b> 00,000	
•	large_range 2000 * (-1010) + [-1000,	✓ <b>OK</b> 500,-1]	
•	extreme_large (-2,, -2, 1,, 1) and (MAX (MAX_INT), length = ~100	,	