

Candidate Report: Anonymous

Test Name:

[Summary](#)[Timeline](#)

Test Score

33 out of 100 points

33%

Tasks in Test

	Time Spent ⓘ	Task Score
MaxProductOfThree Submitted in: Java 8	1 min	33%

TASKS DETAILS

EASY	1. MaxProductOfThree Maximize $A[P] * A[Q] * A[R]$ for any triplet (P, Q, R) .	Task Score 33%	Correctness 0%	Performance ? 60%
------	--	-------------------	-------------------	----------------------

Task description

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

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:

```
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
```

Solution

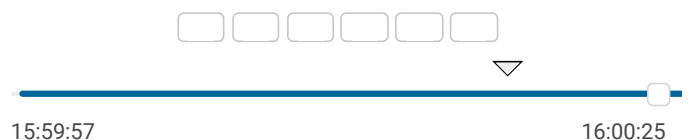
Programming language used: Java 8

Total time used: 1 minutes ⓘ

Effective time used: 1 minutes ⓘ

Notes: *not defined yet*

Task timeline ⓘ



Code: 16:00:25 UTC, java, final,
score: 33

[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 import java.util.Arrays;
8
9 public class Solution {
```

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 2009–2020 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

```
10
11
12     public static int solution(int[] a) {
13         class Wrap implements Comparable<Wrap> {
14             boolean neg;
15             int v;
16
17             @Override
18             public int compareTo(Wrap o) {
19                 return this.v - o.v;
20             }
21         }
22
23         boolean allneg = true;
24         for (int s = 0; s < a.length; s++) {
25             if (a[s] >= 0) {
26                 allneg = false;
27             }
28         }
29
30         if (allneg) {
31             Arrays.sort(a);
32             int r = 1;
33             for (int i = a.length - 1; i > a.l
34                 r = r * a[i];
35
36             return r;
37         }
38
39         Wrap[] w = new Wrap[a.length];
40
41         for (int s = 0; s < a.length; s++) {
42             w[s] = new Wrap();
43             w[s].neg = a[s] < 0;
44             w[s].v = Math.abs(a[s]);
45         }
46
47         Arrays.sort(w);
48
49         int r = 1;
50         int m = 1;
51         for (int i = w.length - 1; i > 0; i--) {
52             m = (w[i].neg ? -1 : 1);
53             r = r * m * w[i].v;
54             if (m == -1 && i <= w.length - 3)
55                 m = 1;
56             r = r / w[i].v;
57         } else if (i <= w.length - 3) {
58             return r;
59         }
60     }
61     return r;
62 }
63
64 }
```

Analysis summary

The following issues have been detected: wrong answers.

For example, for the input [10, 10, 10] the solution returned a wrong answer (got 100 expected 1000).

Analysis ?

Example tests	
▶ example	✓ OK
example test	
Correctness tests	
▶ one_triple	✗ WRONG ANSWER
three elements	
got 100 expected 1000	
▶ simple1	✗ WRONG ANSWER
simple tests	
got -140 expected 105	

Test results - Codility

▶ simple2	✗ WRONG ANSWER
simple tests	got 25 expected 125
▶ small_random	✗ WRONG ANSWER
random small, length = 100	got -964280454 expected 964280454
expand all	Performance tests
▶ medium_range	✗ WRONG ANSWER
-1000, -999, ... 1000, length = ~1,000	got -999000000 expected 999000000
▶ medium_random	✓ OK
random medium, length = ~10,000	
▶ large_random	✓ OK
random large, length = ~100,000	
▶ large_range	✗ WRONG ANSWER
2000 * (-10..10) + [-1000, 500, -1]	got -5000000 expected 5000000
▶ extreme_large	✓ OK
(-2, .., -2, 1, .., 1) and (MAX_INT)..(MAX_INT), length = ~100,000	

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.