Codility_

CodeCheck Report: trainingFKTJM8-TV4

Test Name:

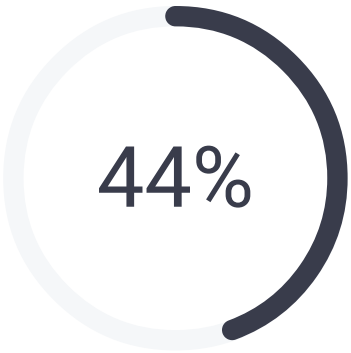
Check out Codility training tasks

Summary Timeline AI Assistant Transcript

Tasks summary

Task	Time spent	Score
MaxProductOfThree C++ ⚠	6 min	44%

Total score



Tasks Details

1. MaxProductOfThree

Easy

Maximize A[P] * A[Q] * A[R] for any triplet (P, Q, R).

Task Score	Correctness	Performance
44%	50%	40%

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 ≤ 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:

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

Solution

Programming language used:	C++
Total time used:	6 minutes
Effective time used:	6 minutes
Notes:	*not defined yet*

Task timeline

04:21:19 04:26:31

Code: 04:26:31 UTC, cpp, final, score: **44** show code in pop-up

```
1  // you can use includes, for example:
2  #include <algorithm>
```

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].


```
3
4  // you can write to stdout for debugging purpo
5  // cout << "this is a debug message" << endl;
6
7  int solution(vector<int> &A) {
8      // (0,1,2) -> A[0]*A[1]*A[2]
9      // find the maximal product of any triplet
10
11     sort(A.begin(), A.end());
12
13     int size = A.size();
14     return A[size-1]*A[size-2]*A[size-3];
15 }
```

Analysis summary

The following issues have been detected: wrong answers.

For example, for the input `[−5, 5, −5, 4]` the solution returned a wrong answer (got -100 expected 125).

Analysis

expand all — **Example tests**

▶ example example test	✔ OK	

expand all — **Correctness tests**

▶ one_triple three elements	✔ OK	
▼ simple1 simple tests	✘ WRONG ANSWER got 84 expected 105	

1. 0.001 s **WRONG ANSWER**, got 84 expected 105
2. 0.001 s **WRONG ANSWER**, got 60 expected 120
3. 0.001 s **OK**
4. 0.001 s **OK**

▼ simple2 simple tests	✘ WRONG ANSWER got -100 expected 125	

1. 0.001 s **WRONG ANSWER**, got -100 expected 125
2. 0.001 s **OK**
3. 0.001 s **OK**

▶ small_random random small, length = 100	✔ OK	

expand all — **Performance tests**

▼ medium_range -1000, -999, ... 1000, length = ~1,000	✘ WRONG ANSWER got 997002000 expected 999000000	

1. 0.001 s **WRONG ANSWER**, got 997002000 expected 999000000

▶ medium_random random medium, length = ~10,000	✔ OK	
▶ large_random random large, length = ~100,000	✔ OK	
▼ large_range 2000 * (-10..10) + [-1000, 500, -1]	✘ WRONG ANSWER got 50000 expected 5000000	

1. 0.001 s **WRONG ANSWER**, got 50000 expected 5000000

▼ extreme_large — ✘ **WRONG ANSWER**

(-2, .., -2, 1, .., 1) and (MAX_INT).. (MAX_INT), length = ~100,000 — got 1 expected 4

1. 0.004 s **WRONG ANSWER**, got 1 expected 4
2. 0.008 s **OK**