Codility_

CodeCheck Report: trainingCQPW2M-99G

Test Name:

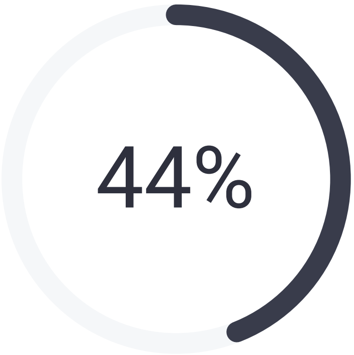
Check out Codility training tasks

Summary Timeline

Tasks summary

Task	Time spent	Score
MaxProductOfThree Python ⚠	23 min	44%

Total score



Tasks Details

		Task Score	Correctness	Performance
Easy	1. MaxProductOfThree Maximize A[P] * A[Q] * A[R] for any triplet (P, Q, R).	44%	100%	0%

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

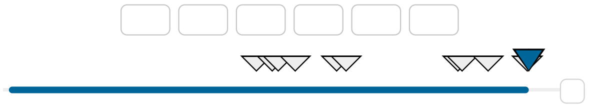
Your goal is to find the maximal product of any triplet.

Write a function:

Solution

Programming language used:	Python
Total time used:	23 minutes ?
Effective time used:	23 minutes ?
Notes:	*not defined yet*

Task timeline ?



07:01:01 07:24:02

Code: 07:24:02 UTC, py, final, score: **44**

show code in pop-up

```
def solution(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].

```
# you can write to stdout for debugging pur
# print("this is a debug message")
from functools import reduce
from itertools import combinations

def solution(A):
    # Implement your solution here
    # pass
    triplets = []
    for triplet in combinations(A, 3):
        triplets.append(triplet)

    triplet_products = []
    for triplet in triplets:
        product = reduce(lambda x, y: x * y
        triplet_products.append(product)

    return max(triplet_products)

```

Analysis summary

The following issues have been detected: timeout errors.

Analysis

Detected time complexity: **O(N**3)**

expand all	Example 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 = 100	✔ OK	
expand all	**Performance tests**		
▶ medium_range	-1000, -999, ... 1000, length = ~1,000	✘ TIMEOUT ERROR	running time: 2.288 sec., time limit: 0.100 sec.
▶ medium_random	random medium, length = ~10,000	✘ TIMEOUT ERROR	running time: 2.344 sec., time limit: 0.100 sec.
▶ large_random	random large, length = ~100,000	✘ TIMEOUT ERROR	running time: 2.320 sec., time limit: 0.240 sec.
▶ large_range	2000 * (-10..10) + [-1000, 500, -1]	✘ TIMEOUT ERROR	running time: 2.296 sec., time limit: 0.112 sec.
▶ extreme_large	(-2, .., -2, 1, .., 1) and (MAX_INT)..(MAX_INT), length = ~100,000	✘ TIMEOUT ERROR	running time: 2.308 sec., time limit: 0.176 sec.