Codility_

Candidate Report: trainingA9MA6Q-EM2

Check out Codility training tasks

Test Name:

Summary Timeline Feedback

Tasks summary

Task	Time spent	Score
MaxProductOfThree C#	1 min	100%

Total score

100%

Tasks Details

Easy

1. MaxProductOfThree	Task Score	Correctness	Performance
Maximize A[P] * A[Q] * A[R] for any triplet (P, Q, R).	100%	100%	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 ≤ 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); }
```

Solution

Programming language used:	C#
Total time used:	1 minutes
Effective time used:	1 minutes
Notes:	*not defined yet*

Task timeline

01:07:57 01:08:30

Code: 01:08:29 UTC, cs, final, score: **100**

show code in pop-up

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].

```
using System;
using System.Linq;
// you can also use other imports, for example:
// using System.Collections.Generic;

// you can write to stdout for debugging purposes, e.g.
// Console.WriteLine("this is a debug message");

class Solution {
   public int solution(int[] A)
       {
           A = A.OrderBy(x => x).ToArray();
           int F = 0;
           int L = A.Length - 1;
           int s1 = A[F] * A[F + 1] * A[F + 2];
           int s2 = A[L] * A[L - 1] * A[F];
           int s3 = A[F] * A[F + 1] * A[L];
           int s4 = A[L] * A[L - 1] * A[L - 2];

           return Math.Max(Math.Max(s1, s2), Math.Max(s3,
       }
}
```

Analysis summary

The solution obtained perfect score.

Analysis

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

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	✓ OK
▶	medium_random random medium, length = ~10,000	✓ OK
▶	large_random random large, length = ~100,000	✓ OK
▶	large_range 2000 * (-10..10) + [-1000, 500, -1]	✓ OK
▶	extreme_large (-2, .., -2, 1, .., 1) and (MAX_INT)..(MAX_INT), length = ~100,000	✓ OK