



Demo ticket

Session

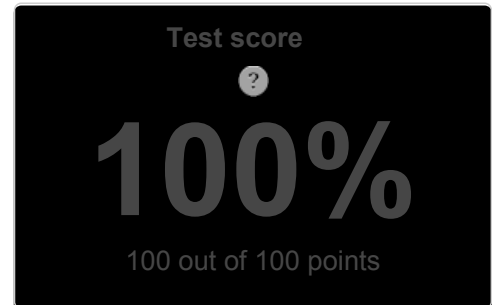
ID: demo52FSF3-Q2V
Time limit: 120 min.

Status: closed

Created on: 2014-03-15 03:05 UTC
Started on: 2014-03-15 03:05 UTC
Finished on: 2014-03-15 03:08 UTC

Tasks in test

Task score



EASY

1. MaxProductOfThree

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

score: 100 of 100



Task description

A non-empty zero-indexed 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:

```
def solution(A)
```

that, given a non-empty zero-indexed 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.

Assume that:

- N is an integer within the range $[3..100,000]$;
- each element of array A is an integer within the range $[-1000..1000]$.

Solution

Programming language used: Python

Total time used: 3 minutes

Effective time used: 3 minutes

Notes: correct functionality and scalability

Task timeline



03:05:49

03:08:07

Code: 03:08:07 UTC, py, final, score: 100.00

```
1. def solution(A):
2.     # write your code in Python 2.6
3.     A.sort()
4.     # because number in A could be nagtive
5.     # neg*neg*pos or pos*pos*pos or neg*neg*neg
6.     return max(A[-1]*A[-2]*A[-3], A[-1]*A[0]*A[1])
```

Analysis

Detected time complexity:

$O(N * \log(N))$

test	time	result
example example test	0.050 s.	OK
one_triple three elements	0.050 s.	OK

[-1,000..1,000].

Complexity:

- expected worst-case time complexity is $O(N \cdot \log(N))$;
- expected worst-case space complexity is $O(1)$, beyond input storage (not counting the storage required for input arguments).

Elements of input arrays can be modified.

Copyright 2009–2014 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Codility

simple1 simple tests	0.050 s.	OK
simple2 simple tests	0.050 s.	OK
small_random random small, length = 100	0.050 s.	OK
medium_range -1000, -999, ... 1000, length = ~1,000	0.050 s.	OK
medium_random random medium, length = ~10,000	0.070 s.	OK
large_random random large, length = ~100,000	0.260 s.	OK
large_range 2000 * (-10..10) + [-1000, 500, -1]	0.130 s.	OK
extreme_large (-2, ..., -2, 1, ..., 1) and (MAX_INT)..(MAX_INT), length = ~100,000	0.210 s.	OK

Training center