|  |  |  |
| --- | --- | --- |
| Name | ID |  |
| Engy ghattas aziz | 20170120 |  |
| Michael esmat Mansour | 20170401 |  |
| Farid faiz farid | 20201857 |  |
| Mohamed khalid abdelhamed | 20201867 |  |
| Mohamed nabil ebrahim | 20180544 |  |

***Pseudo code:***

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

.

class Solution { public int solution(int[] A); }

that, given a non-empty zero-indexed array A, returns the value of the maximal product of any triplet.

Given array A:

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 [−1,000..1,000].

***Complexity:***

* expected worst-case time complexity is O(N\*log(N))

***Code :***

def solution(A):

A.sort()

list\_of\_score = []

if A[0] < 0 and A[1] < 0 and A[-1] > 0:

list\_of\_score.append(A[0] \* A[1] \* A[-1])

list\_of\_score.append(A[-1] \* A[-2] \* A[-3])

list\_of\_score.sort()

return list\_of\_score[-1]

def solution(A):

A.sort()

left = A[0] \* A[1] \* A[-1]

right = A[-1] \* A[-2] \* A[-3]

max\_product = left if left > right else right

return max\_product

Diagram, engineering drawing

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated