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 ≤ 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); }

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 [−1,000..1,000].

Complexity:

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