Problem 3732: Maximum Product of Three Elements After One Replacement

Problem Information

Difficulty: Medium
Acceptance Rate: 46.17%
Paid Only: No
Tags: Array, Math, Greedy, Sorting

Problem Description

You are given an integer array `nums`.

You **must** replace **exactly one** element in the array with **any** integer value in the range `[-105, 105]` (inclusive).

After performing this single replacement, determine the **maximum possible product** of **any three** elements at **distinct indices** from the modified array.

Return an integer denoting the **maximum product** achievable.

Example 1:

Input: nums = [-5,7,0]

Output: 3500000

Explanation:

Replacing 0 with -105 gives the array `[-5, 7, -105]`, which has a product `(-5) * 7 * (-105) = 3500000`. The maximum product is 3500000.

Example 2:

Input: nums = [-4,-2,-1,-3]

Output: 1200000

Explanation:

Two ways to achieve the maximum product include:

* `[-4, -2, -3]` -> replace -2 with 105 -> product = `(-4) * 105 * (-3) = 1200000`. * `[-4, -1, -3]` -> replace -1 with 105 -> product = `(-4) * 105 * (-3) = 1200000`.

The maximum product is 1200000.

Example 3:

Input: nums = [0,10,0]

Output: 0

Explanation:

There is no way to replace an element with another integer and not have a 0 in the array. Hence, the product of all three elements will always be 0, and the maximum product is 0.

Constraints:

* `3 <= nums.length <= 105` * `-105 <= nums[i] <= 105`

Code Snippets

C++:

```
class Solution {
public:
long long maxProduct(vector<int>& nums) {

}
};
```

Java:

```
class Solution {
public long maxProduct(int[] nums) {

}
}
```

Python3:

```
class Solution:
def maxProduct(self, nums: List[int]) -> int:
```