Problem 3670: Maximum Product of Two Integers With No Common Bits

Problem Information

Difficulty: Medium
Acceptance Rate: 12.26%
Paid Only: No
Tags: Array, Dynamic Programming, Bit Manipulation

Problem Description

You are given an integer array `nums`.

Your task is to find two **distinct** indices `i` and `j` such that the product `nums[i] * nums[j]` is **maximized,** and the binary representations of `nums[i]` and `nums[j]` do not share any common set bits.

Return the **maximum** possible product of such a pair. If no such pair exists, return 0.

Example 1:

Input: nums = [1,2,3,4,5,6,7]

Output: 12

Explanation:

The best pair is 3 (011) and 4 (100). They share no set bits and `3 * 4 = 12`.

Example 2:

Input: nums = [5,6,4]

Output: 0

Explanation:

Every pair of numbers has at least one common set bit. Hence, the answer is 0.

Example 3:

Input: nums = [64,8,32]

Output: 2048

Explanation:

No pair of numbers share a common bit, so the answer is the product of the two maximum elements, 64 and 32 (`64 * 32 = 2048`).

Constraints:

* `2 <= nums.length <= 105` * `1 <= nums[i] <= 106`

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:
```