Problem 2708: Maximum Strength of a Group

Problem Information

Difficulty: Medium
Acceptance Rate: 25.22%
Paid Only: No
Tags: Array, Dynamic Programming, Backtracking, Greedy, Bit Manipulation, Sorting, Enumeration

Problem Description

You are given a **0-indexed** integer array `nums` representing the score of students in an exam. The teacher would like to form one **non-empty** group of students with maximal **strength** , where the strength of a group of students of indices `i0`, `i1`, `i2`, ... , `ik` is defined as `nums[i0] * nums[i1] * nums[i2] * ... * nums[ik■]`.

Return _the maximum strength of a group the teacher can create_.

Example 1:

Input: nums = [3,-1,-5,2,5,-9] **Output:** 1350 **Explanation:** One way to form a group of maximal strength is to group the students at indices [0,2,3,4,5]. Their strength is 3 * (-5) * 2 * 5 * (-9) = 1350, which we can show is optimal.

Example 2:

Input: nums = [-4,-5,-4] **Output:** 20 **Explanation:** Group the students at indices [0, 1] . Then, we'll have a resulting strength of 20. We cannot achieve greater strength.

Constraints:

* `1 <= nums.length <= 13` * `-9 <= nums[i] <= 9`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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