Problem 3574: Maximize Subarray GCD Score

Problem Information

Difficulty: Hard
Acceptance Rate: 23.48%
Paid Only: No
Tags: Array, Math, Enumeration, Number Theory

Problem Description

You are given an array of positive integers `nums` and an integer `k`.

You may perform at most `k` operations. In each operation, you can choose one element in the array and **double** its value. Each element can be doubled **at most** once.

The **score** of a contiguous **subarray** is defined as the **product** of its length and the _greatest common divisor (GCD)_ of all its elements.

Your task is to return the **maximum** **score** that can be achieved by selecting a contiguous subarray from the modified array.

Note:

* The **greatest common divisor (GCD)** of an array is the largest integer that evenly divides all the array elements.

Example 1:

Input: nums = [2,4], k = 1

Output: 8

Explanation:

* Double `nums[0]` to 4 using one operation. The modified array becomes `[4, 4]`. * The GCD of the subarray `[4, 4]` is 4, and the length is 2. * Thus, the maximum possible score is `2 × 4

= 8`.

Example 2:

Input: nums = [3,5,7], k = 2

Output: 14

Explanation:

* Double `nums[2]` to 14 using one operation. The modified array becomes `[3, 5, 14]`. * The GCD of the subarray `[14]` is 14, and the length is 1. * Thus, the maximum possible score is `1 × 14 = 14`.

Example 3:

Input: nums = [5,5,5], k = 1

Output: 15

Explanation:

* The subarray `[5, 5, 5]` has a GCD of 5, and its length is 3. * Since doubling any element doesn't improve the score, the maximum score is `3 × 5 = 15`.

Constraints:

* `1 <= n == nums.length <= 1500` * `1 <= nums[i] <= 109` * `1 <= k <= n`

Code Snippets

C++:

```
class Solution {
public:
long long maxGCDScore(vector<int>& nums, int k) {

}
};
```

Java:

```
class Solution {
public long maxGCDScore(int[] nums, int k) {

}
}
```

Python3:

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