Problem 2892: Minimizing Array After Replacing Pairs With Their Product

Problem Information

Difficulty: Medium
Acceptance Rate: 40.55%
Paid Only: Yes
Tags: Array, Dynamic Programming, Greedy

Problem Description

Given an integer array `nums` and an integer `k`, you can perform the following operation on the array any number of times:

* Select two **adjacent** elements of the array like `x` and `y`, such that `x * y <= k`, and replace both of them with a **single element** with value `x * y` (e.g. in one operation the array `[1, 2, 2, 3]` with `k = 5` can become `[1, 4, 3]` or `[2, 2, 3]`, but can't become `[1, 2, 6]`).

Return _the**minimum** possible length of _`nums` _after any number of operations_.

Example 1:

Input: nums = [2,3,3,7,3,5], k = 20 **Output:** 3 **Explanation:** We perform these operations: 1. [_2,3_ ,3,7,3,5] -> [_6_ ,3,7,3,5] 2. [_6,3_ ,7,3,5] -> [_18_ ,7,3,5] 3. [18,7,_3,5_] -> [18,7,_15_] It can be shown that 3 is the minimum length possible to achieve with the given operation.

Example 2:

Input: nums = [3,3,3,3], k = 6 **Output:** 4 **Explanation:** We can't perform any operations since the product of every two adjacent elements is greater than 6. Hence, the answer is 4.

Constraints:

* `1 <= nums.length <= 105` * `0 <= nums[i] <= 109` * `1 <= k <= 109`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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