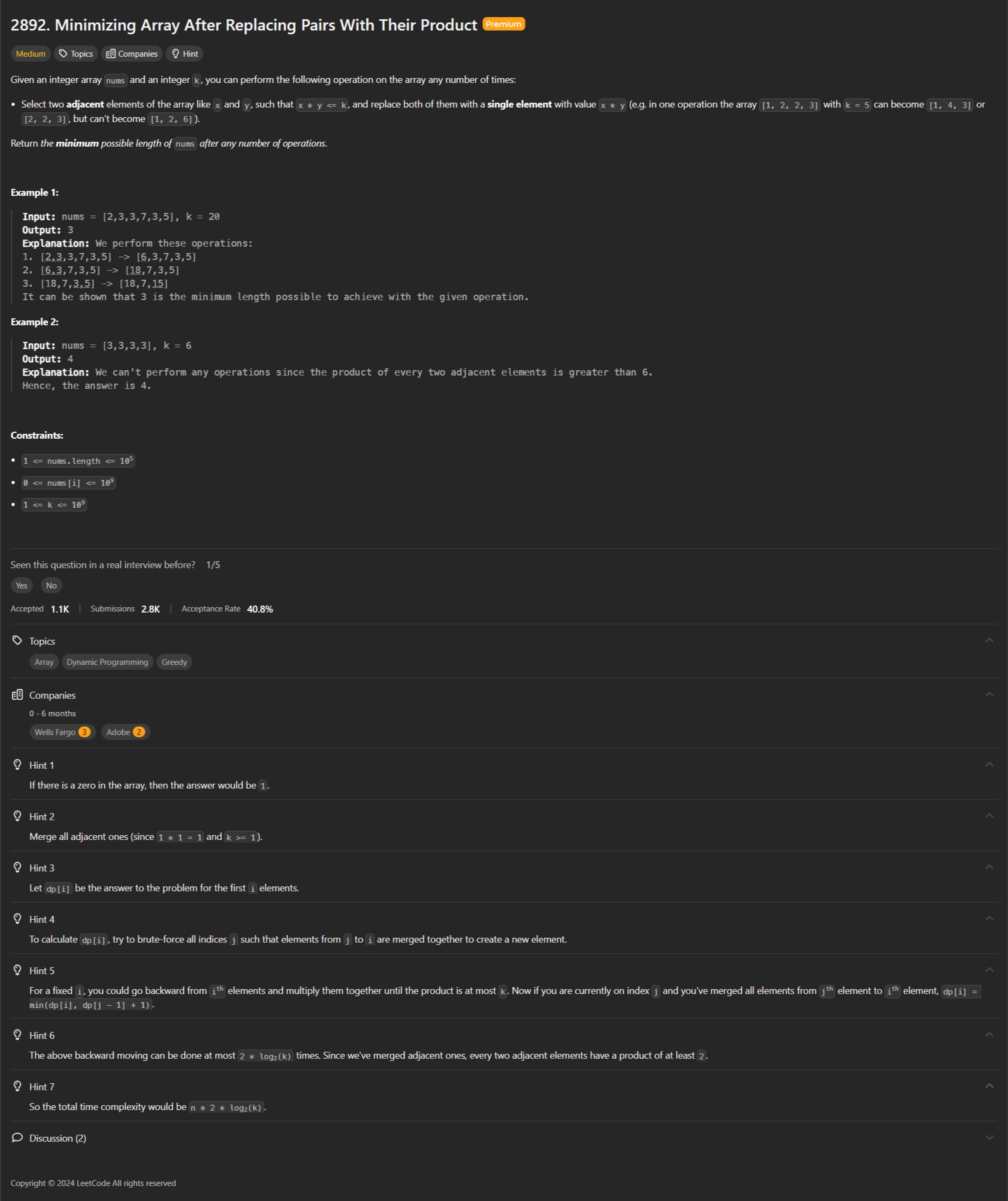

2892. Minimizing Array After Replacing Pairs With Their Product

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:

- $1 <= nums.length <= 10^5$
- $0 <= nums[i] <= 10^9$
- $1 <= k <= 10^9$

Seen this question in a real interview before? 1/5

Accepted 1.1K | Submissions 2.8K | Acceptance Rate 40.8%

Topics

Companies

0 - 6 months

Hint 1

If there is a zero in the array, then the answer would be `1`.

Hint 2

Merge all adjacent ones (since `1 * 1 = 1` and `k >= 1`).

Hint 3

Let `dp[i]` be the answer to the problem for the first `i` elements.

Hint 4

To calculate `dp[i]`, try to brute-force all indices `j` such that elements from `j` to `i` are merged together to create a new element.

Hint 5

For a fixed `i`, you could go backward from i^{th} elements and multiply them together until the product is at most `k`. Now if you are currently on index `j` and you've merged all elements from j^{th} element to i^{th} element, `dp[i] = min(dp[i], dp[j - 1] + 1)`.

Hint 6

The above backward moving can be done at most $2 * \log_2(k)$ times. Since we've merged adjacent ones, every two adjacent elements have a product of at least `2`.

Hint 7

So the total time complexity would be $n * 2 * \log_2(k)$.

Discussion (2)