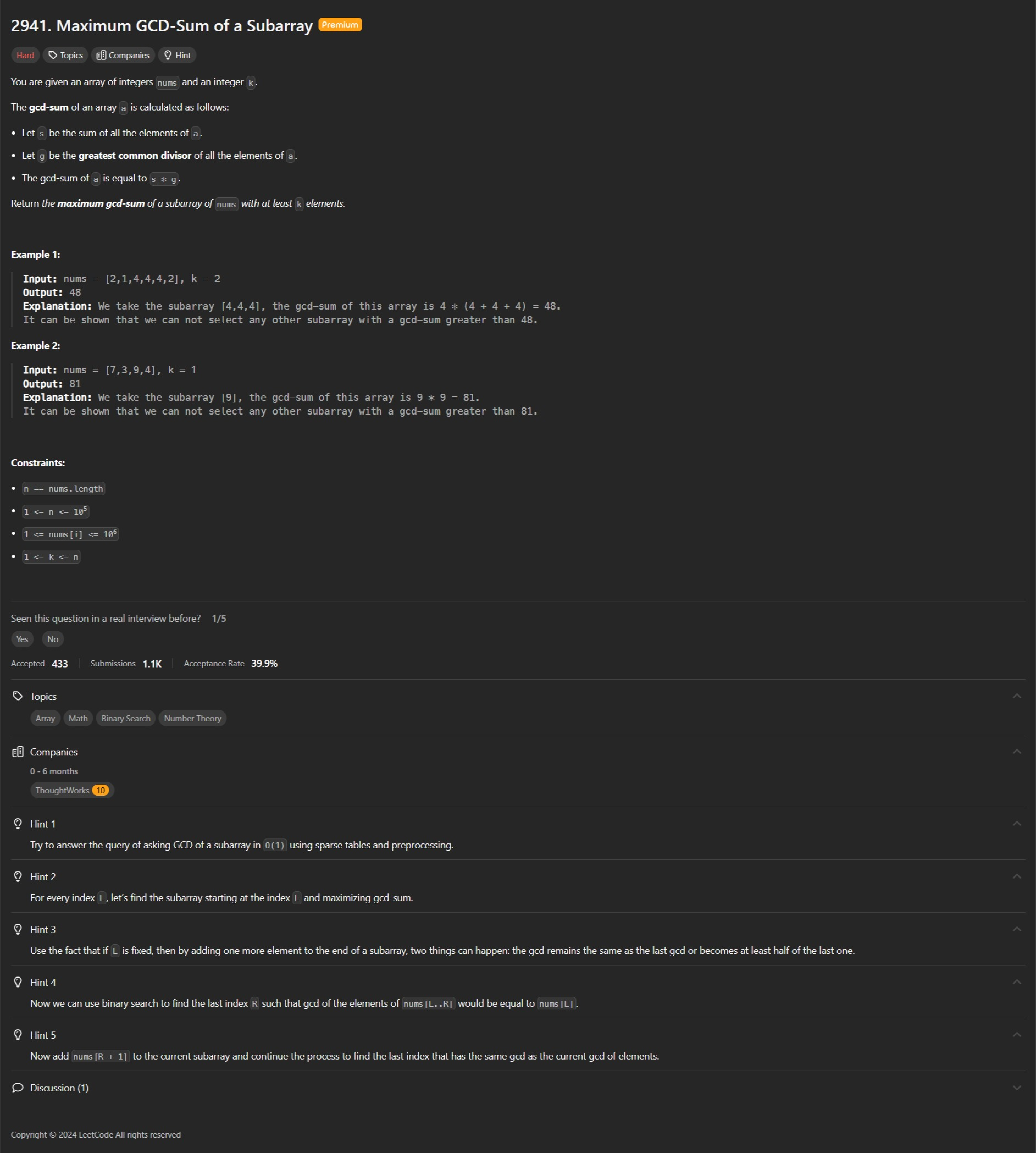

2941. Maximum GCD-Sum of a Subarray

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

The **gcd-sum** of an array `a` is calculated as follows:

- Let `s` be the sum of all the elements of `a`.
- Let `g` be the **greatest common divisor** of all the elements of `a`.
- The gcd-sum of `a` is equal to `s * g`.

Return *the **maximum gcd-sum** of a subarray of* `nums` *with at least* `k` *elements.*

Example 1:

```
Input: nums = [2,1,4,4,4,2], k = 2
Output: 48
Explanation: We take the subarray [4,4,4], the gcd-sum of this array is 4 * (4 + 4 + 4) = 48.
It can be shown that we can not select any other subarray with a gcd-sum greater than 48.
```

Example 2:

```
Input: nums = [7,3,9,4], k = 1
Output: 81
Explanation: We take the subarray [9], the gcd-sum of this array is 9 * 9 = 81.
It can be shown that we can not select any other subarray with a gcd-sum greater than 81.
```

Constraints:

- `n == nums.length`
- $1 <= n <= 10^5$
- $1 <= nums[i] <= 10^6$
- `1 <= k <= n`

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

Accepted 433 | Submissions 1.1K | Acceptance Rate 39.9%

Topics

Array Math Binary Search Number Theory

Companies

0 - 6 months

ThoughtWorks 10

Hint 1

Try to answer the query of asking GCD of a subarray in `O(1)` using sparse tables and preprocessing.

Hint 2

For every index `L`, let's find the subarray starting at the index `L` and maximizing gcd-sum.

Hint 3

Use the fact that if `L` is fixed, then by adding one more element to the end of a subarray, two things can happen: the gcd remains the same as the last gcd or becomes at least half of the last one.

Hint 4

Now we can use binary search to find the last index `R` such that gcd of the elements of `nums[L..R]` would be equal to `nums[L]`.

Hint 5

Now add `nums[R + 1]` to the current subarray and continue the process to find the last index that has the same gcd as the current gcd of elements.

Discussion (1)