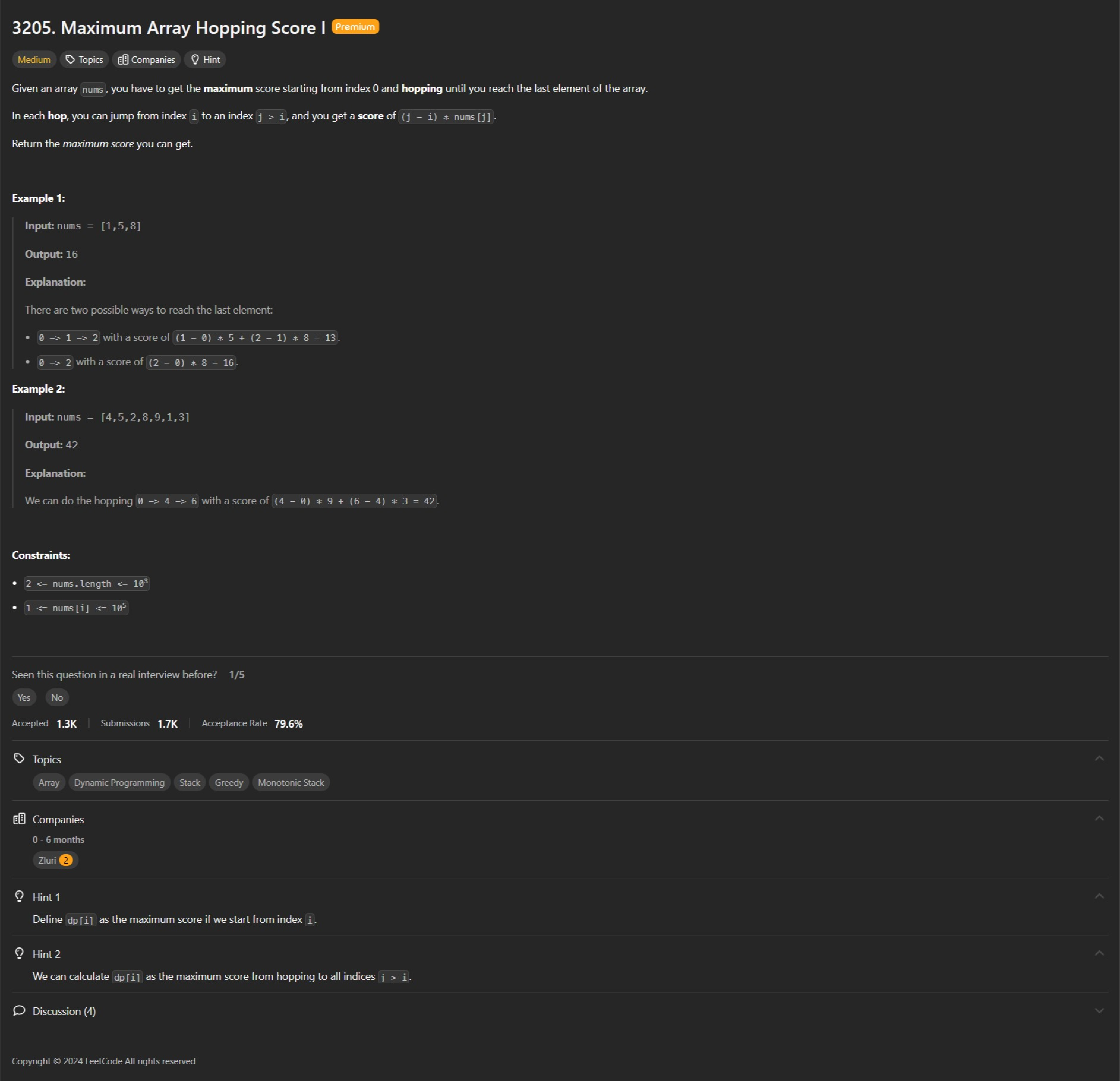

3205. Maximum Array Hopping Score I

Given an array `nums`, you have to get the **maximum** score starting from index 0 and **hopping** until you reach the last element of the array.

In each **hop**, you can jump from index `i` to an index `j > i`, and you get a **score** of `(j - i) * nums[j]`.

Return the *maximum score* you can get.

Example 1:

> **Input:** nums = [1,5,8]
>
> **Output:** 16
>
> **Explanation:**
>
> There are two possible ways to reach the last element:
>
> - `0 -> 1 -> 2` with a score of `(1 - 0) * 5 + (2 - 1) * 8 = 13`.
> - `0 -> 2` with a score of `(2 - 0) * 8 = 16`.

Example 2:

> **Input:** nums = [4,5,2,8,9,1,3]
>
> **Output:** 42
>
> **Explanation:**
>
> We can do the hopping `0 -> 4 -> 6` with a score of `(4 - 0) * 9 + (6 - 4) * 3 = 42`.

Constraints:

- `2 <= nums.length <=` 10^3
- `1 <= nums[i] <=` 10^5

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

Accepted 1.3K | Submissions 1.7K | Acceptance Rate 79.6%

Topics

Companies

0 - 6 months

Hint 1

Define `dp[i]` as the maximum score if we start from index `i`.

Hint 2

We can calculate `dp[i]` as the maximum score from hopping to all indices `j > i`.

Discussion (4)