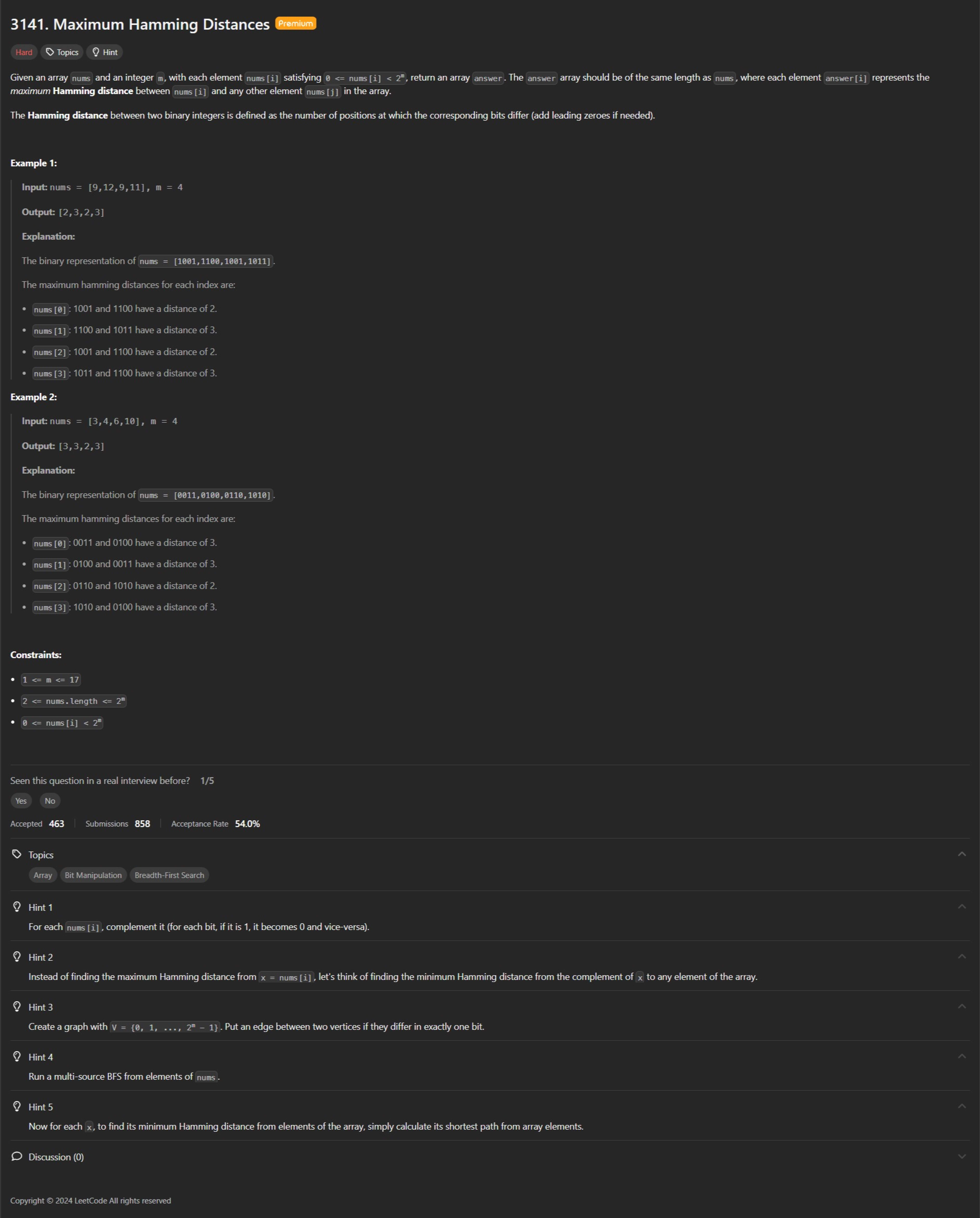

3141. Maximum Hamming Distances

Given an array `nums` and an integer `m`, with each element `nums[i]` satisfying `0 <= nums[i] < 2`^{m}, return an array `answer`. The `answer` array should be of the same length as `nums`, where each element `answer[i]` represents the *maximum* **Hamming distance** between `nums[i]` and any other element `nums[j]` in the array.

The **Hamming distance** between two binary integers is defined as the number of positions at which the corresponding bits differ (add leading zeroes if needed).

Example 1:

> **Input:** `nums = [9,12,9,11], m = 4`
>
> **Output:** `[2,3,2,3]`
>
> **Explanation:**
>
> The binary representation of `nums = [1001,1100,1001,1011]`.
>
> The maximum hamming distances for each index are:
>
> - `nums[0]`: 1001 and 1100 have a distance of 2.
> - `nums[1]`: 1100 and 1011 have a distance of 3.
> - `nums[2]`: 1001 and 1100 have a distance of 2.
> - `nums[3]`: 1011 and 1100 have a distance of 3.

Example 2:

> **Input:** `nums = [3,4,6,10], m = 4`
>
> **Output:** `[3,3,2,3]`
>
> **Explanation:**
>
> The binary representation of `nums = [0011,0100,0110,1010]`.
>
> The maximum hamming distances for each index are:
>
> - `nums[0]`: 0011 and 0100 have a distance of 3.
> - `nums[1]`: 0100 and 0011 have a distance of 3.
> - `nums[2]`: 0110 and 1010 have a distance of 2.
> - `nums[3]`: 1010 and 0100 have a distance of 3.

Constraints:

- `1 <= m <= 17`
- `2 <= nums.length <= 2`^{m}
- `0 <= nums[i] < 2`^{m}

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

Accepted 463 | Submissions 858 | Acceptance Rate 54.0%

Topics

Hint 1

For each `nums[i]`, complement it (for each bit, if it is 1, it becomes 0 and vice-versa).

Hint 2

Instead of finding the maximum Hamming distance from `x = nums[i]`, let's think of finding the minimum Hamming distance from the complement of `x` to any element of the array.

Hint 3

Create a graph with `V = {0, 1, ..., 2`^{m}` - 1}`. Put an edge between two vertices if they differ in exactly one bit.

Hint 4

Run a multi-source BFS from elements of `nums`.

Hint 5

Now for each `x`, to find its minimum Hamming distance from elements of the array, simply calculate its shortest path from array elements.

Discussion (0)