Problem 3488: Closest Equal Element Queries

Problem Information

Difficulty: Medium
Acceptance Rate: 32.26%
Paid Only: No
Tags: Array, Hash Table, Binary Search

Problem Description

You are given a **circular** array `nums` and an array `queries`.

For each query `i`, you have to find the following:

* The **minimum** distance between the element at index `queries[i]` and **any** other index `j` in the **circular** array, where `nums[j] == nums[queries[i]]`. If no such index exists, the answer for that query should be -1.

Return an array `answer` of the **same** size as `queries`, where `answer[i]` represents the result for query `i`.

Example 1:

Input: nums = [1,3,1,4,1,3,2], queries = [0,3,5]

Output: [2,-1,3]

Explanation:

* Query 0: The element at `queries[0] = 0` is `nums[0] = 1`. The nearest index with the same value is 2, and the distance between them is 2. * Query 1: The element at `queries[1] = 3` is `nums[3] = 4`. No other index contains 4, so the result is -1. * Query 2: The element at `queries[2] = 5` is `nums[5] = 3`. The nearest index with the same value is 1, and the distance between them is 3 (following the circular path: `5 -> 6 -> 0 -> 1`).

Example 2:

Input: nums = [1,2,3,4], queries = [0,1,2,3]

Output: [-1,-1,-1,-1]

Explanation:

Each value in `nums` is unique, so no index shares the same value as the queried element. This results in -1 for all queries.

Constraints:

* `1 <= queries.length <= nums.length <= 105` * `1 <= nums[i] <= 106` * `0 <= queries[i] < nums.length`

Code Snippets

C++:

```
class Solution {
public:
vector<int> solveQueries(vector<int>& nums, vector<int>& queries) {

}
};
```

Java:

```
class Solution {
public List<Integer> solveQueries(int[] nums, int[] queries) {

}
}
```

Python3:

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