Problem 3629: Minimum Jumps to Reach End via Prime Teleportation

Problem Information

Difficulty: Medium
Acceptance Rate: 26.82%
Paid Only: No
Tags: Array, Hash Table, Math, Breadth-First Search, Number Theory

Problem Description

You are given an integer array `nums` of length `n`.

You start at index 0, and your goal is to reach index `n - 1`.

From any index `i`, you may perform one of the following operations:

* **Adjacent Step** : Jump to index `i + 1` or `i - 1`, if the index is within bounds. * **Prime Teleportation** : If `nums[i]` is a prime number `p`, you may instantly jump to any index `j != i` such that `nums[j] % p == 0`.

Return the **minimum** number of jumps required to reach index `n - 1`.

Example 1:

Input: nums = [1,2,4,6]

Output: 2

Explanation:

One optimal sequence of jumps is:

* Start at index `i = 0`. Take an adjacent step to index 1. * At index `i = 1`, `nums[1] = 2` is a prime number. Therefore, we teleport to index `i = 3` as `nums[3] = 6` is divisible by 2.

Thus, the answer is 2.

Example 2:

Input: nums = [2,3,4,7,9]

Output: 2

Explanation:

One optimal sequence of jumps is:

* Start at index `i = 0`. Take an adjacent step to index `i = 1`. * At index `i = 1`, `nums[1] = 3` is a prime number. Therefore, we teleport to index `i = 4` since `nums[4] = 9` is divisible by 3.

Thus, the answer is 2.

Example 3:

Input: nums = [4,6,5,8]

Output: 3

Explanation:

* Since no teleportation is possible, we move through `0 -> 1 -> 2 -> 3`. Thus, the answer is 3.

Constraints:

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

Code Snippets

C++:

```
class Solution {
public:
int minJumps(vector<int>& nums) {
}
};
```

Java:

```
class Solution {
public int minJumps(int[] nums) {

}
}
```

Python3:

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