Problem 3202: Find the Maximum Length of Valid Subsequence II

Problem Information

Difficulty: Medium
Acceptance Rate: 57.27%
Paid Only: No
Tags: Array, Dynamic Programming

Problem Description

You are given an integer array `nums` and a **positive** integer `k`.

A subsequence `sub` of `nums` with length `x` is called **valid** if it satisfies:

* `(sub[0] + sub[1]) % k == (sub[1] + sub[2]) % k == ... == (sub[x - 2] + sub[x - 1]) % k.`

Return the length of the **longest** **valid** subsequence of `nums`.

Example 1:

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

Output: 5

Explanation:

The longest valid subsequence is `[1, 2, 3, 4, 5]`.

Example 2:

Input: nums = [1,4,2,3,1,4], k = 3

Output: 4

Explanation:

The longest valid subsequence is `[1, 4, 1, 4]`.

Constraints:

* `2 <= nums.length <= 103` * `1 <= nums[i] <= 107` * `1 <= k <= 103`

Code Snippets

C++:

```
class Solution {
public:
int maximumLength(vector<int>& nums, int k) {

}
};
```

Java:

```
class Solution {
public int maximumLength(int[] nums, int k) {

}
}
```

Python3:

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