Problem 3434: Maximum Frequency After Subarray Operation

Problem Information

Difficulty: Medium
Acceptance Rate: 30.16%
Paid Only: No
Tags: Array, Hash Table, Dynamic Programming, Greedy, Enumeration, Prefix Sum

Problem Description

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

You perform the following operation on `nums` **once** :

* Select a subarray `nums[i..j]` where `0 <= i <= j <= n - 1`. * Select an integer `x` and add `x` to **all** the elements in `nums[i..j]`.

Find the **maximum** frequency of the value `k` after the operation.

Example 1:

Input: nums = [1,2,3,4,5,6], k = 1

Output: 2

Explanation:

After adding -5 to `nums[2..5]`, 1 has a frequency of 2 in `[1, 2, -2, -1, 0, 1]`.

Example 2:

Input: nums = [10,2,3,4,5,5,4,3,2,2], k = 10

Output: 4

Explanation:

After adding 8 to `nums[1..9]`, 10 has a frequency of 4 in `[10, 10, 11, 12, 13, 13, 12, 11, 10, 10]`.

Constraints:

* `1 <= n == nums.length <= 105` * `1 <= nums[i] <= 50` * `1 <= k <= 50`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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