Problem 2025: Maximum Number of Ways to Partition an Array

Problem Information

Difficulty: Hard
Acceptance Rate: 35.12%
Paid Only: No
Tags: Array, Hash Table, Counting, Enumeration, Prefix Sum

Problem Description

You are given a **0-indexed** integer array `nums` of length `n`. The number of ways to **partition** `nums` is the number of `pivot` indices that satisfy both conditions:

* `1 <= pivot < n` * `nums[0] + nums[1] + ... + nums[pivot - 1] == nums[pivot] + nums[pivot + 1] + ... + nums[n - 1]`

You are also given an integer `k`. You can choose to change the value of **one** element of `nums` to `k`, or to leave the array **unchanged**.

Return _the**maximum** possible number of ways to **partition** _`nums` _to satisfy both conditions after changing**at most** one element_.

Example 1:

Input: nums = [2,-1,2], k = 3 **Output:** 1 **Explanation:** One optimal approach is to change nums[0] to k. The array becomes [**_3_** ,-1,2]. There is one way to partition the array: - For pivot = 2, we have the partition [3,-1 | 2]: 3 + -1 == 2.

Example 2:

Input: nums = [0,0,0], k = 1 **Output:** 2 **Explanation:** The optimal approach is to leave the array unchanged. There are two ways to partition the array: - For pivot = 1, we have the partition [0 | 0,0]: 0 == 0 + 0. - For pivot = 2, we have the partition [0,0 | 0]: 0 + 0 == 0.

Example 3:

Input: nums = [22,4,-25,-20,-15,15,-16,7,19,-10,0,-13,-14], k = -33 **Output:** 4 **Explanation:** One optimal approach is to change nums[2] to k. The array becomes [22,4,_**-33**_ ,-20,-15,15,-16,7,19,-10,0,-13,-14]. There are four ways to partition the array.

Constraints:

* `n == nums.length` * `2 <= n <= 105` * `-105 <= k, nums[i] <= 105`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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