Problem 3077: Maximum Strength of K Disjoint Subarrays

Problem Information

Difficulty: Hard
Acceptance Rate: 27.29%
Paid Only: No
Tags: Array, Dynamic Programming, Prefix Sum

Problem Description

You are given an array of integers `nums` with length `n`, and a positive **odd** integer `k`.

Select exactly **`k`** disjoint subarrays **`sub 1, sub2, ..., subk`** from `nums` such that the last element of `subi` appears before the first element of `sub{i+1}` for all `1 <= i <= k-1`. The goal is to maximize their combined strength.

The strength of the selected subarrays is defined as:

`strength = k * sum(sub1)- (k - 1) * sum(sub2) + (k - 2) * sum(sub3) - ... - 2 * sum(sub{k-1}) + sum(subk)`

where **`sum(sub i)`** is the sum of the elements in the `i`-th subarray.

Return the **maximum** possible strength that can be obtained from selecting exactly **`k`** disjoint subarrays from `nums`.

Note that the chosen subarrays **don 't** need to cover the entire array.

Example 1:

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

Output: 22

Explanation:

The best possible way to select 3 subarrays is: nums[0..2], nums[3..3], and nums[4..4]. The strength is calculated as follows:

`strength = 3 * (1 + 2 + 3) - 2 * (-1) + 2 = 22`

Example 2:

Input: nums = [12,-2,-2,-2,-2], k = 5

Output: 64

Explanation:

The only possible way to select 5 disjoint subarrays is: nums[0..0], nums[1..1], nums[2..2], nums[3..3], and nums[4..4]. The strength is calculated as follows:

`strength = 5 * 12 - 4 * (-2) + 3 * (-2) - 2 * (-2) + (-2) = 64`

Example 3:

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

Output: -1

Explanation:

The best possible way to select 1 subarray is: nums[0..0]. The strength is -1.

Constraints:

* `1 <= n <= 104` * `-109 <= nums[i] <= 109` * `1 <= k <= n` * `1 <= n * k <= 106` * `k` is odd.

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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