Problem 3430: Maximum and Minimum Sums of at Most Size K Subarrays

Problem Information

Difficulty: Hard
Acceptance Rate: 23.56%
Paid Only: No
Tags: Array, Math, Stack, Monotonic Stack

Problem Description

You are given an integer array `nums` and a **positive** integer `k`. Return the sum of the **maximum** and **minimum** elements of all subarrays with **at most** `k` elements.

Example 1:

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

Output: 20

Explanation:

The subarrays of `nums` with at most 2 elements are:

Subarray | Minimum | Maximum | Sum ---|---|---|--- `[1]` | 1 | 1 | 2 `[2]` | 2 | 2 | 4 `[3]` | 3 | 3 | 6 `[1, 2]` | 1 | 2 | 3 `[2, 3]` | 2 | 3 | 5 **Final Total** | | | 20 The output would be 20.

Example 2:

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

Output: -6

Explanation:

The subarrays of `nums` with at most 2 elements are:

Subarray | Minimum | Maximum | Sum ---|---|---|--- `[1]` | 1 | 1 | 2 `[-3]` | -3 | -3 | -6 `[1]` | 1 | 1 | 2 `[1, -3]` | -3 | 1 | -2 `[-3, 1]` | -3 | 1 | -2 **Final Total** | | | -6 The output would be -6.

Constraints:

* `1 <= nums.length <= 80000` * `1 <= k <= nums.length` * `-106 <= nums[i] <= 106`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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