Problem 1749: Maximum Absolute Sum of Any Subarray

Problem Information

Difficulty: Medium
Acceptance Rate: 71.39%
Paid Only: No
Tags: Array, Dynamic Programming

Problem Description

You are given an integer array `nums`. The **absolute sum** of a subarray `[numsl, numsl+1, ..., numsr-1, numsr]` is `abs(numsl + numsl+1 + ... + numsr-1 + numsr)`.

Return _the**maximum** absolute sum of any **(possibly empty)** subarray of _`nums`.

Note that `abs(x)` is defined as follows:

* If `x` is a negative integer, then `abs(x) = -x`. * If `x` is a non-negative integer, then `abs(x) = x`.

Example 1:

Input: nums = [1,-3,2,3,-4] **Output:** 5 **Explanation:** The subarray [2,3] has absolute sum = abs(2+3) = abs(5) = 5.

Example 2:

Input: nums = [2,-5,1,-4,3,-2] **Output:** 8 **Explanation:** The subarray [-5,1,-4] has absolute sum = abs(-5+1-4) = abs(-8) = 8.

Constraints:

* `1 <= nums.length <= 105` * `-104 <= nums[i] <= 104`

Code Snippets

C++:

```
class Solution {
public:
int maxAbsoluteSum(vector<int>& nums) {

}
};
```

Java:

```
class Solution {
public int maxAbsoluteSum(int[] nums) {

}
}
```

Python3:

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