1749. Maximum Absolute Sum of Any Subarray

Medium | Topics | Companies | Hint

You are given an integer array `nums`. The **absolute sum** of a subarray `[nums_l, nums_{l+1}, ..., nums_{r-1}, nums_r]` is `abs(nums_l + nums_{l+1} + ... + nums_{r-1} + nums_r)`.

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.
```
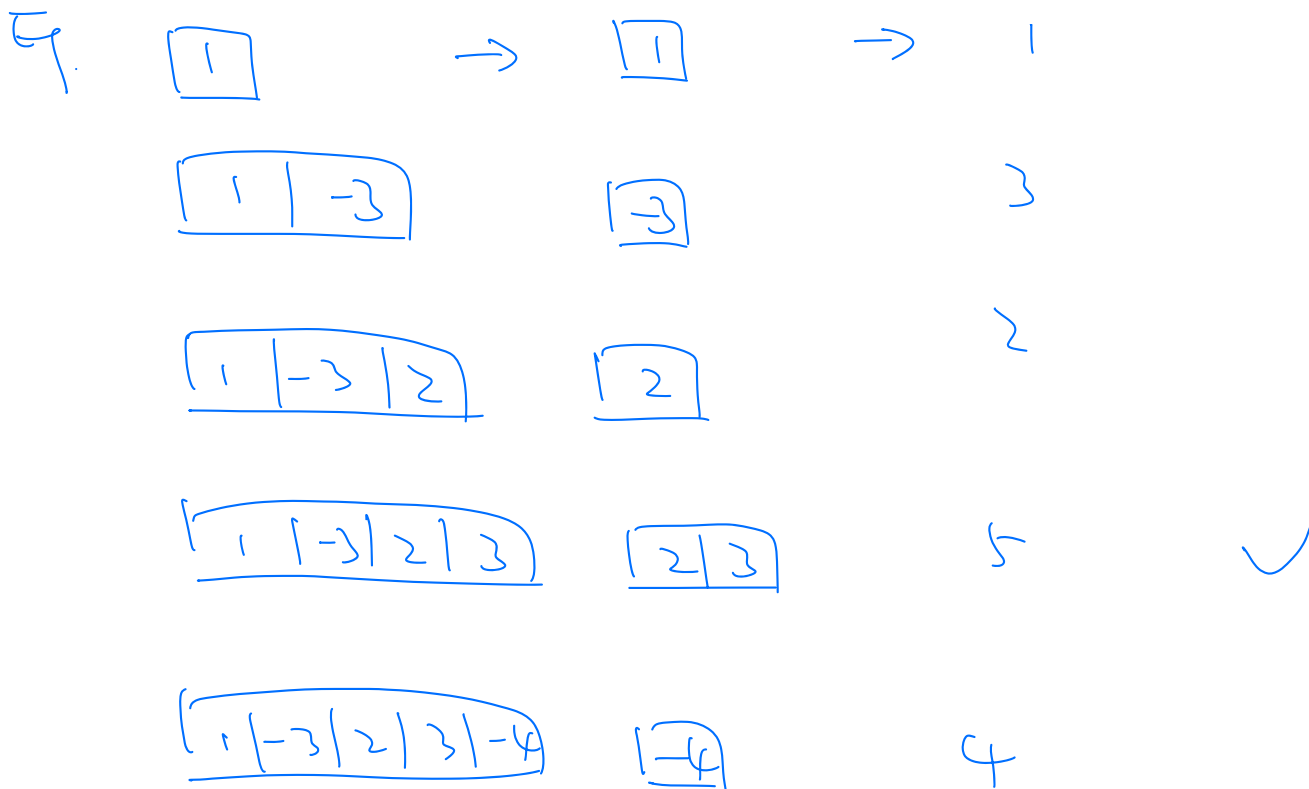
check
Max Subarray Sum

1, -3, 2, 3, -4

idea is find the max abs sub ending at
index i.

$\rightarrow$ give fixed $\boxed{|\ \cdots\ |}$, remove prefix

$\underline{|\!\!\!\!\diagdown\diagdown\diagdown\diagdown\!\!\!\cdot\cdot|}\ )$, to get curr_max

notice :

removing prefix is repeating work.

so, find a pattern

$\rightarrow$ maintaing a prefix_sum array

nums: 1, -3, 2, 3, -4

prefix_sum: 1, -2, 0, 3, -1

each time end at i.

curr = nums[i]

we vars to maintain this
$$\begin{cases} curr\_pre\_min = min\ (\ prefix\_sum\ [:i]) \\ curr\_pre\_max = max\ (\qquad \cdot \qquad \cdot \qquad) \end{cases}$$

temp = max(abs (curr - curr_pre_min),

abs (curr - curr_pre_max)

)

Accepted 66 / 66 testcases passed

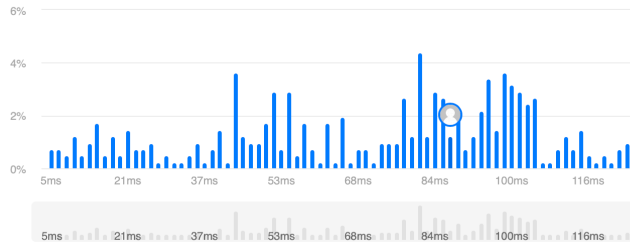AndrewC275 submitted at Feb 26, 2025 14:49

Editorial Solution

🕐 Runtime ⓘ

87 ms | Beats **42.82%**

✦ Analyze Complexity

⚙ Memory

28.50 MB | Beats **63.99%** 🌱

6%

4%

2%

0%

5ms 21ms 37ms 53ms 68ms 84ms 100ms 116ms

5ms 21ms 37ms 53ms 68ms 84ms 100ms 116ms

```python
class Solution:
    def maxAbsoluteSum(self, nums: List[int]) -> int:
        res = 0
        curr = 0
        prefix_min = 0
        prefix_max = 0

        for n in nums:
            curr += n

            res = max(res, abs(curr - prefix_min), abs(curr - prefix_max))

            prefix_min = min(prefix_min, curr)
            prefix_max = max(prefix_max, curr)

        return res
```

Ln 11, Col 62 | Saved

1° res

2° prefix_min/max is for next curr