2036. Maximum Alternating Subarray Sum `Premium`

`Medium` | 🏷 Topics | 🏢 Companies | 💡 Hint

A **subarray** of a **0-indexed** integer array is a contiguous **non-empty** sequence of elements within an array.

The **alternating subarray sum** of a subarray that ranges from index `i` to `j` (**inclusive**, `0 <= i <= j < nums.length`) is `nums[i] - nums[i+1] + nums[i+2] - ... +/- nums[j]`.

Given a **0-indexed** integer array `nums`, return *the **maximum alternating subarray sum** of any subarray of* `nums`.

Example 1:

```
Input: nums = [3,-1,1,2]
Output: 5
Explanation:
The subarray [3,-1,1] has the largest alternating subarray sum.
The alternating subarray sum is 3 - (-1) + 1 = 5.
```

Example 2:

```
Input: nums = [2,2,2,2,2]
Output: 2
Explanation:
The subarrays [2], [2,2,2], and [2,2,2,2,2] have the largest alternating subarray sum.
The alternating subarray sum of [2] is 2.
The alternating subarray sum of [2,2,2] is 2 - 2 + 2 = 2.
The alternating subarray sum of [2,2,2,2,2] is 2 - 2 + 2 - 2 + 2 = 2.
```

Example 3:

```
Input: nums = [1]
Output: 1
Explanation:
There is only one non-empty subarray, which is [1].
The alternating subarray sum is 1.
```

Constraints:

- $1 <= nums.length <= 10^5$
- $-10^5 <= nums[i] <= 10^5$

Seen this question in a real interview before?   1/5

Yes   No

Accepted **3.4K** | Submissions **8.5K** | Acceptance Rate **40.0%**

🏷 Topics   ⌃

Array   Dynamic Programming

🏢 Companies   ⌃

0 - 6 months

Amazon 2

💡 Hint 1   ⌃

How can Kadane's Algorithm help us?

💡 Hint 2   ⌃

If you convert all the numbers at odd indices to the negative version of that number, the problem simplifies to finding the maximum subarray sum.

💡 Hint 3   ⌃

However, this strategy needs you to start each subarray at an even index.

💡 Hint 4   ⌃

Do the same except converting all the numbers at even indices to the negative version of that number.

🗎 Similar Questions

Maximum Alternating Subsequence Sum   Medium

💬 Discussion (1)   ⌃