5782. Maximum Alternating Subsequence Sum

My Submissions (/contest/biweekly-contest-55/problems/maximum-alternating-subsequence-sum/submissions/)

Back to Contest (/contest/biweekly-contest-55/)

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Medium

The **alternating sum** of a **0-indexed** array is defined as the **sum** of the elements at **even** indices **minus** the **sum** of the elements at **odd** indices.

- For example, the alternating sum of `[4,2,5,3]` is `(4 + 5) - (2 + 3) = 4`.

Given an array `nums`, return *the **maximum alternating sum** of any subsequence of* `nums` *(after **reindexing** the elements of the subsequence)*.

A **subsequence** of an array is a new array generated from the original array by deleting some elements (possibly none) without changing the remaining elements' relative order. For example, `[2,7,4]` is a subsequence of `[4,2,3,7,2,1,4]` (the underlined elements), while `[2,4,2]` is not.

Example 1:

```
Input: nums = [4,2,5,3]
Output: 7
Explanation: It is optimal to choose the subsequence [4,2,5] with alternating sum (4 + 5) - 2 = 7.
```

Example 2:

```
Input: nums = [5,6,7,8]
Output: 8
Explanation: It is optimal to choose the subsequence [8] with alternating sum 8.
```

Example 3:

```
Input: nums = [6,2,1,2,4,5]
Output: 10
Explanation: It is optimal to choose the subsequence [6,1,5] with alternating sum (6 + 5) - 1 = 10.
```

Constraints:

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

JavaScript

```
/**
 * @param {number[]} nums
 * @return {number}
 */
var maxAlternatingSum = function(nums) {

};
```

Custom Testcase Use Example Testcases

Run Submit