5709. Maximum Ascending Subarray Sum

My Submissions (/contest/weekly-contest-233/problems/maximum-ascending-subarray-sum/submissions/) Back to Contest (/contest/weekly-contest-233/)

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

Given an array of positive integers `nums`, return the *maximum possible sum of an* ***ascending*** *subarray in* `nums`.

A subarray is defined as a contiguous sequence of numbers in an array.

A subarray `[nums`$_l$`, nums`$_{l+1}$`, ..., nums`$_{r-1}$`, nums`$_r$`]` is **ascending** if for all `i` where `l <= i < r`, `nums`$_i$ `<` `nums`$_{i+1}$. Note that a subarray of size `1` is **ascending**.

Example 1:

```
Input: nums = [10,20,30,5,10,50]
Output: 65
Explanation: [5,10,50] is the ascending subarray with the maximum sum of 65.
```

Example 2:

```
Input: nums = [10,20,30,40,50]
Output: 150
Explanation: [10,20,30,40,50] is the ascending subarray with the maximum sum of 150.
```

Example 3:

```
Input: nums = [12,17,15,13,10,11,12]
Output: 33
Explanation: [10,11,12] is the ascending subarray with the maximum sum of 33.
```

Example 4:

```
Input: nums = [100,10,1]
Output: 100
```

Constraints:

- `1 <= nums.length <= 100`
- `1 <= nums[i] <= 100`

JavaScript

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

};
```