

Description

Editorial

Solutions (760)

Submissions

## 1800. Maximum Ascending Subarray Sum

Hint

⋮

Easy

👍 586

💬 19

☆

🔄

🏢 Companies

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 `[numsl, numsl+1, ..., numsr-1, numsr]` is **ascending** if for all `i` where `l <= i < r`, `numsi < numsi+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.

### Constraints:

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

Accepted **41.5K** | Submissions **65.7K** | Acceptance Rate **63.1%**

Discussion (1)

⌵

Similar Questions

⌵

Related Topics

⌵