1746. Maximum Subarray Sum After One Operation Premium

Medium | Topics | Companies | Hint

You are given an integer array `nums`. You must perform **exactly one** operation where you can **replace** one element `nums[i]` with `nums[i] * nums[i]`.

Return *the **maximum** possible subarray sum after **exactly one** operation*. The subarray must be non-empty.

Example 1:

```
Input: nums = [2,-1,-4,-3]
Output: 17
Explanation: You can perform the operation on index 2 (0-indexed) to make nums = [2,-1,16,-3]. Now, the maximum subarray sum is 2 + -1 + 16 = 17.
```

Example 2:

```
Input: nums = [1,-1,1,1,-1,-1,1]
Output: 4
Explanation: You can perform the operation on index 1 (0-indexed) to make nums = [1,1,1,1,-1,-1,1]. Now, the maximum subarray sum is 1 + 1 + 1 + 1 = 4.
```

Constraints:

- $1 \le nums.length \le 10^5$
- $-10^4 \le nums[i] \le 10^4$

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

Yes No

Accepted 9.4K | Submissions 15.2K | Acceptance Rate 62.0%

Topics

Array Dynamic Programming

Companies

0 - 6 months

Sprinklr 2

Hint 1

Think about dynamic programming

Hint 2

Define an array dp[nums.length][2], where dp[i][0] is the max subarray sum including nums[i] and without squaring any element.

Hint 3

dp[i][1] is the max subarray sum including nums[i] and having only one element squared.

Similar Questions

Maximum Subarray Medium

Discussion (3)