Problem 1856: Maximum Subarray Min-Product

Problem Information

Difficulty: Medium
Acceptance Rate: 39.55%
Paid Only: No
Tags: Array, Stack, Monotonic Stack, Prefix Sum

Problem Description

The **min-product** of an array is equal to the **minimum value** in the array **multiplied by** the array's **sum**.

* For example, the array `[3,2,5]` (minimum value is `2`) has a min-product of `2 * (3+2+5) = 2 * 10 = 20`.

Given an array of integers `nums`, return _the**maximum min-product** of any **non-empty subarray** of _`nums`. Since the answer may be large, return it **modulo** `109 + 7`.

Note that the min-product should be maximized **before** performing the modulo operation. Testcases are generated such that the maximum min-product **without** modulo will fit in a **64-bit signed integer**.

A **subarray** is a **contiguous** part of an array.

Example 1:

Input: nums = [1,_2,3,2_] **Output:** 14 **Explanation:** The maximum min-product is achieved with the subarray [2,3,2] (minimum value is 2). 2 * (2+3+2) = 2 * 7 = 14.

Example 2:

Input: nums = [2,_3,3_ ,1,2] **Output:** 18 **Explanation:** The maximum min-product is achieved with the subarray [3,3] (minimum value is 3). 3 * (3+3) = 3 * 6 = 18.

Example 3:

Input: nums = [3,1,_5,6,4_ ,2] **Output:** 60 **Explanation:** The maximum min-product is achieved with the subarray [5,6,4] (minimum value is 4). 4 * (5+6+4) = 4 * 15 = 60.

Constraints:

* `1 <= nums.length <= 105` * `1 <= nums[i] <= 107`

Code Snippets

C++:

```
class Solution {
public:
int maxSumMinProduct(vector<int>& nums) {

}
};
```

Java:

```
class Solution {
public int maxSumMinProduct(int[] nums) {

}
}
```

Python3:

```
class Solution:
def maxSumMinProduct(self, nums: List[int]) -> int:
```