Problem 2702: Minimum Operations to Make Numbers Non-positive

Problem Information

Difficulty: Hard
Acceptance Rate: 43.25%
Paid Only: Yes
Tags: Array, Binary Search

Problem Description

You are given a **0-indexed** integer array `nums` and two integers `x` and `y`. In one operation, you must choose an index `i` such that `0 <= i < nums.length` and perform the following:

* Decrement `nums[i]` by `x`. * Decrement values by `y` at all indices except the `ith` one.

Return _the minimum number of operations to make all the integers in_`nums` _**less than or equal to zero.**_

Example 1:

Input: nums = [3,4,1,7,6], x = 4, y = 2 **Output:** 3 **Explanation:** You will need three operations. One of the optimal sequence of operations is: Operation 1: Choose i = 3. Then, nums = [1,2,-1,3,4]. Operation 2: Choose i = 3. Then, nums = [-1,0,-3,-1,2]. Operation 3: Choose i = 4. Then, nums = [-3,-2,-5,-3,-2]. Now, all the numbers in nums are non-positive. Therefore, we return 3.

Example 2:

Input: nums = [1,2,1], x = 2, y = 1 **Output:** 1 **Explanation:** We can perform the operation once on i = 1. Then, nums becomes [0,0,0]. All the positive numbers are removed, and therefore, we return 1.

Constraints:

* `1 <= nums.length <= 105` * `1 <= nums[i] <= 109` * `1 <= y < x <= 109`

Code Snippets

C++:

```
class Solution {
public:
int minOperations(vector<int>& nums, int x, int y) {

}
};
```

Java:

```
class Solution {
public int minOperations(int[] nums, int x, int y) {

}
}
```

Python3:

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