Problem 360: Sort Transformed Array

Problem Information

Difficulty: Medium
Acceptance Rate: 57.55%
Paid Only: Yes
Tags: Array, Math, Two Pointers, Sorting

Problem Description

Given a **sorted** integer array `nums` and three integers `a`, `b` and `c`, apply a quadratic function of the form `f(x) = ax2 + bx + c` to each element `nums[i]` in the array, and return _the array in a sorted order_.

Example 1:

Input: nums = [-4,-2,2,4], a = 1, b = 3, c = 5 **Output:** [3,9,15,33]

Example 2:

Input: nums = [-4,-2,2,4], a = -1, b = 3, c = 5 **Output:** [-23,-5,1,7]

Constraints:

* `1 <= nums.length <= 200` * `-100 <= nums[i], a, b, c <= 100` * `nums` is sorted in **ascending** order.

Follow up: Could you solve it in `O(n)` time?

Code Snippets

C++:

```
class Solution {
public:
```

```
vector<int> sortTransformedArray(vector<int>& nums, int a, int b, int c) {

}
};
```

Java:

```
class Solution {
public int[] sortTransformedArray(int[] nums, int a, int b, int c) {

}
}
```

Python3:

```
class Solution:
def sortTransformedArray(self, nums: List[int], a: int, b: int, c: int) ->
List[int]:
```