Array #3

Leetcode #977

Squares of a Sorted Array

https://leetcode.com/problems/squares-of-a-sorted-array/description/

Given an integer array nums sorted in non-decreasing order, return an array of the squares of each number sorted in non-ascending order

decreasing

Example 1:
Input: nums = [-4, -1, 0, 3, 10]
Output: [0, 1, 9, 16, 100]

16 1 0 9 100

0 1 9 16 100

Example 2:
Input: nums = [-7, -3, 2, 3, 11]
Output: [4, 9, 9, 49, 121]

Constraints:
1 <= nums.length <= 10^4
-10^4 <= nums[i] <= 10^4
nums is sorted in non-decreasing order

Companies:
Meta, Yandex, Google, Amazon, Apple, Yahoo, etc

Approach 1:
Replace each element by its square
Sort it

10 -5 1 2 4 7

nums 2 100 25 1 4 16 49

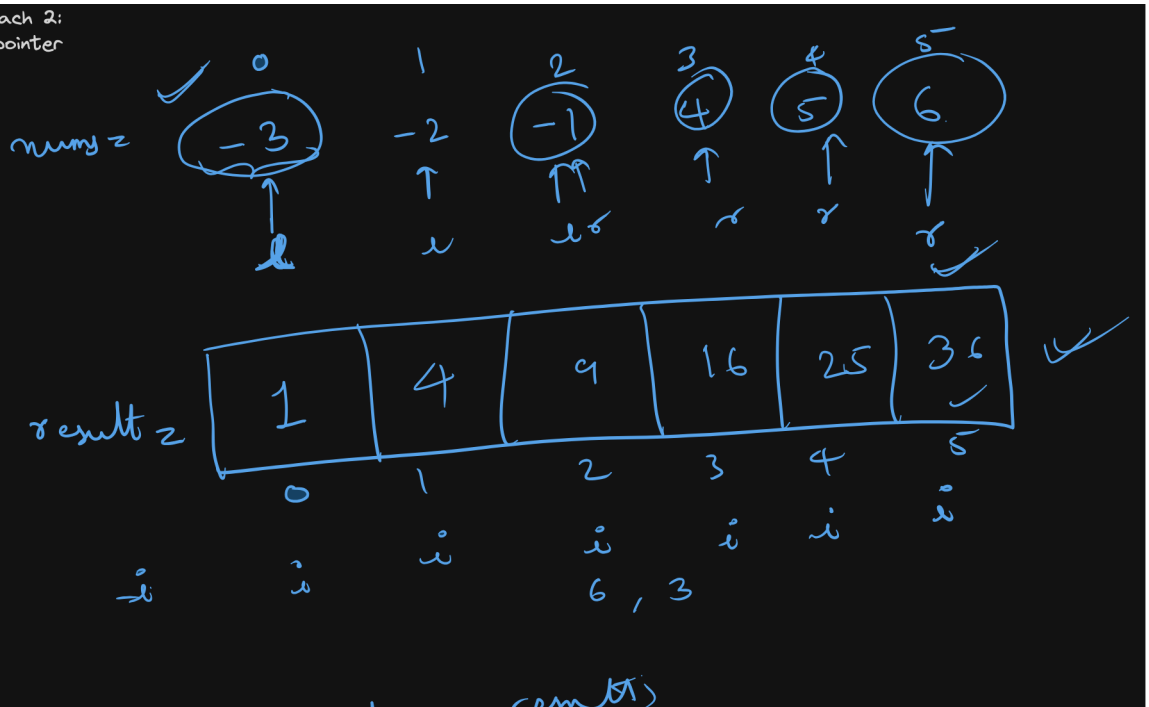
9

i

sort => 1 4 9 16 25 100

Time complexity:
O(N.logN)
Space Complexity:
O(logN)

Approach 2:
Two pointer



return result;

Time complexity:
O(N)
Space Complexity:
O(N)

```
class Solution {
    public int[] sortedSquares(int[] nums) {
        int N = nums.length;
        int[] result = new int[N];

        for (int i = 0; i < N; i++) {
            result[i] = nums[i] * nums[i];
        }

        Arrays.sort(result);

        return result;
    }
}
```

```
class Solution {
    public int[] sortedSquares(int[] nums) {
        int n = nums.length;
        int l = 0,
            r = nums.length - 1;

        int[] result = new int[n];

        for (int i = n - 1; i >= 0; i--) {
            int val;

            if (Math.abs(nums[l]) > Math.abs(nums[r])) {
                val = nums[l];
                l++;
            } else {
                val = nums[r];
```

```
            }

            result[i] = val * val;
        }

        return result;
    }
}
```