**Squares of a Sorted Array**

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

<https://leetcode.com/explore/featured/card/fun-with-arrays/521/introduction/3240/>

**Example 1:**

**Input:** nums = [-4,-1,0,3,10]

**Output:** [0,1,9,16,100]

**Explanation:** After squaring, the array becomes [16,1,0,9,100].

After sorting, it becomes [0,1,9,16,100].

**Example 2:**

**Input:** nums = [-7,-3,2,3,11]

**Output:** [4,9,9,49,121]

**Constraints:**

* 1 <= nums.length <= 104
* -104 <= nums[i] <= 104
* nums is sorted in **non-decreasing** order.

**Follow up:** Squaring each element and sorting the new array is very trivial, could you find an O(n) solution using a different approach?

**Solution :**

**class Solution {**

**public:**

**vector<int> sortedSquares(vector<int>& nums) {**

**int n=nums.size();**

**for(int i=0; i<n; i++){**

**nums[i]=nums[i]\*nums[i];**

**}**

**sort(nums.begin(),nums.end());**

**return nums;**

**}**

**};**