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

Ex:

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

output: [0, 1, 9, 16, 100]

Constraints:

-> 1 <= nums.length <= 10^4

-> -10^4 <= nums[i] <= 10^4

Algorithm:

1) Input a sorted array nums

2) Iterate through elements of array and replace each element with its square

nums[i] *= nums[i].

3) sort the array using bubble sort

-> compare each pair of elements and swap if left one is bigger than right one

-> keep repeating until whole array is sorted

4) Return the sorted array.

Time complexity : $O(n^2)$

space complexity : $O(1)$

code:

```
for(int i=0; i<nums.length ; i++){
    nums[i] *=nums[i]; // replace element with its square
}
// sort using Bubblesort
for(int i=0; i<nums.length-1; i++){ // each pass move
                                    largest element to corr
                                    pla
    for(int j=0; j<nums.length-1-i ; j++){
        // swap if left is bigger than right
        int temp=nums[j];
        nums[j]=nums[j+1];
        nums[j+1]=temp;
    }
}
return nums; // returns the sorted array.
```