

## 977. Squares of a Sorted Array

Solved

Easy Topics Companies

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

### 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.

The screenshot shows a code editor interface with the following components:

- Top Bar:** Navigation links (Description, Accepted, Editorial, Solutions, Submissions), a 'Run' button, a 'Submit' button, and a 'Premium' badge.
- Left Panel:**
  - Accepted:** A message indicating the solution was submitted on Jul 03, 2024 at 02:06.
  - Runtime:** 142 ms, Beats 93.11%.
  - Memory:** 63.26 MB, Beats 45.30%.
  - Complexity Graph:** A bar chart showing the distribution of runtime and memory usage across various test cases.
  - Code:** A snippet of C# code for the solution.
- Right Panel:**
  - Code:** The full C# solution code for the 'SortedSquares' method.
  - Testcase:** A section showing the test results for 'Case 1' and 'Case 2', both of which are 'Accepted'.
  - Test Result:** A detailed view of the test results, showing the input array `[-4,-1,0,3,10]` and the expected output `[0,1,9,16,100]`.

Код:

```
public class Solution
{
    public int[] SortedSquares(int[] nums)
    {
        int n = nums.Length;
```

```
int[] result = new int[n];
int left = 0;
int right = n - 1;
int pos = n - 1;

while (left <= right)
{
    int leftSquare = nums[left] * nums[left];
    int rightSquare = nums[right] * nums[right];

    if (leftSquare > rightSquare)
    {
        result[pos] = leftSquare;
        left++;
    }
    else
    {
        result[pos] = rightSquare;
        right--;
    }
    pos--;
}

return result;
}
```