Merge Sorted Array

Question: https://leetcode.com/problems/merge-sorted-array/

You are given two integer arrays `nums1` and `nums2`, sorted in non-decreasing order, and two integers `m` and `n`, representing the number of elements in `nums1` and `nums2` respectively.

Merge `nums1` and `nums2` into a single array sorted in non-decreasing order.

The final sorted array should not be returned by the function, but instead, be stored inside the array `nums1`. To accommodate this, `nums1` has a length of `m + n`, where the first `m` elements denote the elements that should be merged, and the last `n` elements are set to `0` and should be ignored. `nums2` has a length of `n`.

Example 1:

Input: nums1 = [1,2,3,0,0,0], m = 3, nums2 = [2,5,6], n = 3
Output: [1,2,2,3,5,6]
Explanation: The arrays we are merging are [1,2,3] and [2,5,6].
The result of the merge is [1,2,2,3,5,6] with the underlined elements coming from nums1.

This was a pretty straightforward question and the name itself suggests that it is involving Merge Sort.

Merge Sort Explanation: https://www.geeksforgeeks.org/merge-sort/

Since the question asked for changing **nums1** to return the answer, therefore I used another array **nums3** to store a copy of this array that I used for sort comparison and made the final changes in **nums1**

My Solution:

```
def merge(self, nums1: List[int], m: int, nums2: List[int], n: int) -> None:
    """
    Do not return anything, modify nums1 in-place instead.
    """
    nums3=nums1[:]
    i,j,k=0,0,0
    while(i<m and j<n):
        if(nums3[i]<=nums2[j]):
            nums1[k]=nums3[i]
            i+=1
        else:
            nums1[k]=nums2[j]
            j+=1
        k+=1
    while(i<m):
        nums1[k]=nums3[i]
        i+=1
        k+=1
    while(j<n):
        nums1[k]=nums2[j]
        j+=1
        k+=1
```

A simple merge sort implementation.

Code: https://gist.github.com/vermaayush680/5eba9500568e87ef7b1941647f97ee3a

Time Complexity: O(m+n)
Space Complexity: O(m)