Merge Sorted Array

Problem condition:

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**.

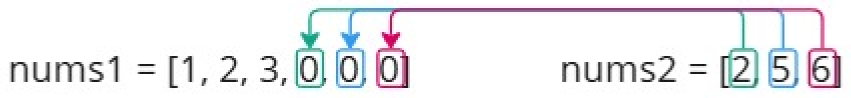
The final sorted array should not be returned by the function, but instead be *stored inside the array* nums1. To accommodate this, **nums1 has a length of m + n**, where the first m elements denote the elements that should be merged, and **the last n elements are set to 0** and should be ignored. nums2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

Algorithm:

Just merge arrays and then sort the resulting array

Example nums1 = [1, 2, 3, 0, 0, 0], m = 3, nums2 = [2, 5, 6], n = 3:

At first we need to "move" all elements of nums2 array to the last n element of nums1 array



nums1 = [1, 2, 3, 0, 0, 0]   nums2 = [2, 5, 6]

After merge nums1 = [1, 2, 3, 2, 5, 6]

Then we can use python built-in method sort() to sort array nums1

After sorting nums1 = [1, 2, 2, 3, 5, 6]