**88. Merge Sorted Array**

Easy

7331644Add to ListShare

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.

class Solution {

public void merge(int[] nums1, int m, int[] nums2, int n) {

int i=m-1;

int j=n-1;

int index=nums1.length-1;

while(i>=0 && j>=0)

{

int l=nums1[i];

int r=nums2[j];

if(l<r)

{

nums1[index--]=r;

j--;

}

else

{

nums1[index--]=l;

i--;

}

}

while (j>=0)

{

int r=nums2[j];

nums1[index--]=r;

j--;

}

}

}

