88 合并两个有序数组

Lable：双指针
给你两个有序整数数组 nums1 和 nums2，请你将 nums2 合并到 nums1 中，使 nums1 成为一个有序数组。
说明：
    初始化 nums1 和 nums2 的元素数量分别为 m 和 n 。
    你可以假设 nums1 有足够的空间（空间大小大于或等于 m + n）来保存 nums2 中的元素。
示例：
    输入：
    nums1 = [1,2,3,0,0,0], m = 3
    nums2 = [2,5,6],   n = 3
    输出：[1,2,2,3,5,6]

- 先合并，再排序

```
// 源 nums2    0 从源的第0个元素开始复制
// 目标  nums1  从mums1的第m个位置开始添加  复制n个元素
class Solution {
  public void merge(int[] nums1, int m, int[] nums2, int n) {
    System.arraycopy(nums2, 0, nums1, m, n);
    Arrays.sort(nums1);
  }
}
```

- 双指针，从前往后

```
class Solution {
  public void merge(int[] nums1, int m, int[] nums2, int n) {
    // Make a copy of nums1.
    int [] nums1_copy = new int[m];
    System.arraycopy(nums1, 0, nums1_copy, 0, m);

    // Two get pointers for nums1_copy and nums2.
    int p1 = 0;
    int p2 = 0;

    // Set pointer for nums1
    int p = 0;

    // 向 nums1中添加最小元素，直至某一个数组添加完
    while ((p1 < m) && (p2 < n))
      nums1[p++] = (nums1_copy[p1] < nums2[p2]) ? nums1_copy[p1++] :
nums2[p2++];

    // 填充剩余未比较的元素
    if (p1 < m)
      System.arraycopy(nums1_copy, p1, nums1, p1 + p2, m + n - p1 - p2);
    if (p2 < n)
      System.arraycopy(nums2, p2, nums1, p1 + p2, m + n - p1 - p2);
  }
}
```