**难度: 简单**

给定两个排序后的数组 A 和 B，其中 A 的末端有足够的缓冲空间容纳 B。 编写一个方法，将 B 合并入 A 并排序。

初始化 A 和 B 的元素数量分别为 m 和 n。

示例:

输入:

A = [1,2,3,0,0,0], m = 3

B = [2,5,6], n = 3

输出: [1,2,2,3,5,6]

说明:

A.length == n + m

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/sorted-merge-lcci

著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。

**思路1:**

1). 直接把B数组里面的数全部添加在A数组的最后

2). 最后直接排序

|  |
| --- |
| public void merge(int[] A, int m, int[] B, int n) {  int length = m + n;  int index = 0;  for (int i = m; i < length; i++) {  A[i] = B[index++];  }  Arrays.sort(A);  } |

**思路2:**

1). A的有效数组的值和B有效数组的值进行比较,谁大就把哪个值从A数组最后一个开始插入

2). 如果第一个数组是空,直接将数组B所有的值,插入到A就行了

|  |
| --- |
| public void merge(int[] A, int m, int[] B, int n) {  //最终的大小  int length = m + n;  //-1是因为下标是从0开始, 如果直接length就越界了  for (int i = length -1; i >= 0; i--) {  //两个数组都有值  if (m > 0 && n > 0) {  if (A[m -1] < B[n-1] ) {  A[i] = B[--n];  }else {  A[i] = A[--m];  }  }else if (n > 0 && m == 0) {  //第一个数组的值为0  A[i] = B[--n];  }  }  } |