

合并两个有序数组

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]

解题思路：

牺牲空间，创建额外一个数组进行最终顺序合并数组存放

首先同时遍历数组 1 和数组 2

数组 1 和数组 2 中元素比较，谁小谁先放在数组 3 中

注意：

可能数组 1 和数组 2 的长度并不一致，在遍历比较过程中，可能一方率先结束，此时，剩余部分一定是大于之前所有数组中元素，所以，剩余部分可以直接拷贝到数组 3 当前结尾

代码如下：

```
void merge(int* nums1, int nums1Size, int m, int* nums2, int nums2Size, int n){
    int* nums3=(int*)malloc(sizeof(int)*(m+n));
    //同时遍历
    int idx=0;
    int i=0;
    int j=0;
    while(i<m&& j<n)
    {
        if(nums1[i]<=nums2[j])
        {
            nums3[idx++]=nums1[i++];
        }
        else
```

```
        {
            nums3[idx++] = nums2[j++];
        }
    }
    // 剩余因为有序可直接拷贝
    if (i < m)
    {
        memcpy(nums3 + idx, nums1 + i, sizeof(int) * (m - i));
    }
    if (j < n)
    {
        memcpy(nums3 + idx, nums2 + j, sizeof(int) * (n - j));
    }
    memcpy(nums1, nums3, sizeof(int) * (m + n));
}
```