

88-合并两个有序数组

题述

88. 合并两个有序数组

难度 简单

1070

☆ 收藏

🔗 分享

🌐 切换为英文

🔔 接收动态

🗉 反馈

给你两个按 **非递减顺序** 排列的整数数组 `nums1` 和 `nums2`，另有两个整数 `m` 和 `n`，分别表示 `nums1` 和 `nums2` 中的元素数目。

请你 **合并** `nums2` 到 `nums1` 中，使合并后的数组同样按 **非递减顺序** 排列。

注意：最终，合并后数组不应由函数返回，而是存储在数组 `nums1` 中。为了应对这种情况，`nums1` 的初始长度为 `m + n`，其中前 `m` 个元素表示应合并的元素，后 `n` 个元素为 0，应忽略。`nums2` 的长度为 `n`。

示例 1：

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]`
解释：需要合并 `[1,2,3]` 和 `[2,5,6]`。
合并结果是 `[1,2,2,3,5,6]`，其中斜体加粗标注的为 `nums1` 中的元素。

示例 2：

输入：`nums1 = [1]`, `m = 1`, `nums2 = []`, `n = 0`
输出：`[1]`
解释：需要合并 `[1]` 和 `[]`。
合并结果是 `[1]`。

示例 3：

输入：`nums1 = [0]`, `m = 0`, `nums2 = [1]`, `n = 1`
输出：`[1]`
解释：需要合并的数组是 `[]` 和 `[1]`。
合并结果是 `[1]`。
注意，因为 `m = 0`，所以 `nums1` 中没有元素。`nums1` 中仅存的 0 仅仅是为了确保合并结果可以顺利存放到 `nums1` 中。

浅析

真没啥好说的

直接想好逻辑顺着往下写就行了

至于题目里的非递减顺序

直接给合并后的vector排个序就行了嘛

代码

```
class Solution {
```

```

public:
    void merge(vector<int>& nums1, int m, vector<int>& nums2, int n)
    {
        //这道题
        //简单的给我整不会了
        //时间复杂度为o(n)  n为传入的第二个数组的大小
        int k=0;
        for(int i=m;i<m+n;i++)
        {
            nums1[i]=nums2[k];
            k++;
        }
        sort(nums1.begin(),nums1.end());
    }
};

```

AC

执行结果: **通过** [显示详情](#)

[添加备注](#)

执行用时: **0 ms**, 在所有 C++ 提交中击败了 **100.00%** 的用户

内存消耗: **8.8 MB**, 在所有 C++ 提交中击败了 **70.18%** 的用户

通过测试用例: **59 / 59**

炫耀一下:



[写题解, 分享我的解题思路](#)

提交结果	执行用时	内存消耗	语言	提交时间	备注
通过	0 ms	8.8 MB	C++	2021/08/28 15:18	添加备注
通过	0 ms	8.9 MB	C++	2021/08/28 15:05	添加备注
通过	4 ms	8.9 MB	C++	2021/08/28 15:03	添加备注