

0088. 合并两个有序数组

👤 [ITCharge](#) ⌚ 大约 2 分钟

- 标签：数组、双指针、排序
- 难度：简单

题目链接

- [0088. 合并两个有序数组 - 力扣](#)

题目大意

描述： 给定两个有序数组 $nums1$ 、 $nums2$ 。

要求： 将 $nums2$ 合并到 $nums1$ 中，使 $nums1$ 成为一个有序数组。

说明：

- 给定数组 $nums1$ 空间大小为 $m + n$ 个，其中前 m 个为 $nums1$ 的元素。 $nums2$ 空间大小为 n 。这样可以用 $nums1$ 的空间来存储最终的有序数组。
- $nums1.length == m + n$ 。
- $nums2.length == n$ 。
- $0 \leq m, n \leq 200$ 。
- $1 \leq m + n \leq 200$ 。
- $-10^9 \leq nums1[i], nums2[j] \leq 10^9$ 。

示例：

- 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。

py

- 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]`。

解题思路

思路 1: 快慢指针

1. 将两个指针 `index1`、`index2` 分别指向 `nums1`、`nums2` 数组的尾部, 再用一个指针 `index` 指向数组 `nums1` 的尾部。
2. 从后向前判断当前指针下 `nums1[index1]` 和 `nums2[index2]` 的值大小, 将较大值存入 `nums1[index]` 中, 然后继续向前遍历。
3. 最后再将 `nums2` 中剩余元素赋值到 `nums1` 前面对应位置上。

思路 1: 代码

```
class Solution:
    def merge(self, nums1: List[int], m: int, nums2: List[int], n: int) -> None:
        index1 = m - 1
        index2 = n - 1
        index = m + n - 1
        while index1 >= 0 and index2 >= 0:
            if nums1[index1] < nums2[index2]:
                nums1[index] = nums2[index2]
                index2 -= 1
            else:
                nums1[index] = nums1[index1]
                index1 -= 1
            index -= 1

        nums1[:index2+1] = nums2[:index2+1]
```

思路 1: 复杂度分析

- 时间复杂度: $O(m + n)$ 。

- 空间复杂度: $O(m + n)$ 。