

Problem

88. Merge Sorted Array(Easy)

Given two sorted integer arrays `nums1` and `nums2`, merge `nums2` into `nums1` as one sorted array.

Note:

- The number of elements initialized in `nums1` and `nums2` are `m` and `n` respectively.
- You may assume that `nums1` has enough space (size that is greater or equal to `m + n`) to hold additional elements from `nums2`.

Example:

Input:

`nums1 = [1,2,3,0,0,0]`, `m = 3`

`nums2 = [2,5,6]`, `n = 3`

Output: `[1,2,2,3,5,6]`

Solution

$O(M+N)$ time, $O(1)$ space

对于有序数组的合并，立刻能想到的是**归并排序**，归并排序通常是**从前向后**扫描合并，对于链表而言，插入是 $O(1)$ 的，然而数组不是。由于题目时将结果存在`num1`中，当`num1s[i]<nums2[j]`，我们需要将`nums2[j]`插入`nums1`中，为了避免对有序数组做插入操作，最直观的想法是用`nums1[i]`存入一个队列中，然而这种方法需要额外的存储空间；那这时想到`num2[j]`之前的空间可以存储`num1[i]`，这样节省了存储空间，但是实现较为麻烦。

可以换个思路，**从后向前**扫描合并，第一次比较的结果插入`nums1[m+n-1]`，这样的话

- 当`j<0`时，合并结束
- 当`i<0`时，`nums2`中所有剩余的元素按顺序插入即可

[GitHub传送门](#)

```
class Solution
{
public:
    void merge(vector<int> &nums1, int m, vector<int> &nums2, int n)
    {
        int k = m + n - 1, i = m - 1, j = n - 1;
        while (k >= 0)
        {
            if (j < 0)
                break;
            if (i < 0)
                nums1[k--] = nums2[j--];
            else
                nums1[k--] = nums1[i] > nums2[j] ? nums1[i--] : nums2[j--];
        }
    }
};
```