



DAILY PROGRAMMING CHALLENGE



Merge Two Sorted Arrays

You are given two sorted arrays `arr1` of size `m` and `arr2` of size `n`. Your task is to merge these two arrays into a single sorted array without using any extra space (i.e., in-place merging). The elements in `arr1` should be merged first, followed by the elements of `arr2`, resulting in both arrays being sorted after the merge.

Input:

Two sorted integer arrays `arr1` of size `m` and `arr2` of size `n`.

Example :

`arr1 = [1, 3, 5, 7]`

`arr2 = [2, 4, 6, 8]`

Output:

Both `arr1` and `arr2` should be sorted after the merge. Since you cannot use extra space, the final result will be reflected in `arr1` and `arr2`.

Example:

`arr1 = [1, 2, 3, 4]`

`arr2 = [5, 6, 7, 8]`

Constraints:

- The arrays are sorted in non-decreasing order.
- You must not use any extra space beyond a few variables ($O(1)$ space complexity).
- $1 \leq m, n \leq 10^5$.
- $1 \leq arr1[i], arr2[j] \leq 10^9$.

Test Cases:

1. Test Case 1

Input: `arr1 = [1, 3, 5]`, `arr2 = [2, 4, 6]`

Output: `arr1 = [1, 2, 3]`, `arr2 = [4, 5, 6]`

2. Test Case 2:

Input: `arr1 = [10, 12, 14]`, `arr2 = [1, 3, 5]`

Output: `arr1 = [1, 3, 5]`, `arr2 = [10, 12, 14]`



3. Test Case 3:

Input: arr1 = [2, 3, 8], arr2 = [4, 6, 10]

Output: arr1 = [2, 3, 4], arr2 = [6, 8, 10]

4. Test Case 4:

Input: arr1 = [1], arr2 = [2]

Output: arr1 = [1], arr2 = [2]

5. Test Case 5:

Input: arr1 = [1, 2, 3, 4, ..., 100000], arr2 = [50001, ..., 100000]

Output: arr1 = [1, 2, 3, ..., 50000], arr2 = [50001, ..., 100000]

Edge Cases:

1. One or both arrays are already sorted in such a way that no swaps are needed.
2. One array is significantly smaller than the other.