Merge Two Sorted Arrays Without Extra Space - Notes

Problem: You are given two sorted arrays arr1 and arr2. The task is to merge them into sorted form without using extra space. That means you cannot use an additional array for merging. Constraints: - Both arrays are already sorted individually. - The merged result should remain sorted across arr1 and arr2. - Only in-place swapping and sorting operations are allowed.

Approach: 1. Start from the end of arr1 and the beginning of arr2. 2. Compare elements: - If arr1[i] > arr2[j], swap them and move i-- and j++. - Otherwise, break (arrays are already sorted in relative order). 3. After the loop, sort arr1 and arr2 individually to maintain complete ordering. 4. This ensures the final sequence across both arrays is sorted. Time Complexity: O((n + m) log(n + m)) due to sorting. Space Complexity: O(1) (in-place operations only).

```
void mergeTwoSortedArraysWithoutExtraSpace(vector<long long> &arr1, vector<long long> &arr2)
{
    int n = arr1.size(), m = arr2.size();
    int i = n - 1, j = 0;
    while (i >= 0 && j < m)
    {
        if (arr1[i] > arr2[j])
        {
            swap(arr1[i], arr2[j]);
            i--;
            j++;
        }
        else
        {
            break;
        }
    }
    sort(arr1.begin(), arr1.end());
    sort(arr2.begin(), arr2.end());
}
```