MERGE SORTED ARRAYS

☆ In this problem, we are supposed to merge two sorted arrays without using any extra space. We will not be creating an extra array for the answers. We will replace the items within the first and second arrays.

So if array1 has l elements and if array2 has m elements, first l elements of the merged will be in array1 and rest of them in array2.

There are only two optimal solutions.

In the first one, we are supposed to start from the last element of the first array and first element of the last array. We start by checking if array2[0] < array1[N-1] that means array2[0] should be on the left array. We swap. We keep checking and as soon as it stops happening, we stop swapping.

Now both the arrays are unsorted so we sort them and done.

Pseudocode :

mergeSortedArrays(arr1, int m, arr2, int n){

```
    int left = m - 1
    int right = 0

    while (left >= 0 && right <= n) {
        if (arr1[left] > arr2[right]) {
            swap(arr1[left], arr2[right])
        } else {
            break
        }
    }
    sort(arr1)
    sort(arr2)
}
```