**Move Zeroes**

rearranging an array in-place by moving all zeros to the end while maintaining the relative order of the non-zero elements.

1. **Problem solving:**

**Two-Pointer Approach:**

Utilize a two-pointer approach for an in-place solution without using additional space.

One pointer (nonZeroIndex) keeps track of the position where the next non-zero element should be placed.

The other pointer traverses the array to find non-zero elements.

**Move Non-Zero Elements to the Front:**

Iterate through the array with the second pointer.

If the element is non-zero, place it at the position indicated by the nonZeroIndex pointer and increment the nonZeroIndex.

**Fill Remaining Elements with Zeros:**

After moving all non-zero elements to the front, the nonZeroIndex points to the first position where zeros should be placed.

Iterate from nonZeroIndex to the end of the array and fill these positions with zeros.

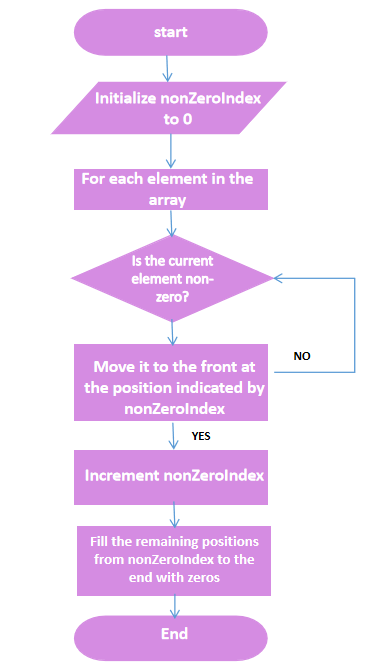
**Maintain Relative Order:**

The algorithm ensures that the relative order of non-zero elements is maintained because they are moved to the front in the order they appear in the array.

**In-Place Modification:**

The algorithm modifies the input array in-place without creating a copy, meeting the requirement of the problem.

1. **Flow chart**



**End**

1. **Output** 