MOVE ZEROS TO END

In this question, all non zero elements are to be shifted to the start and all zeroes are to be moved to the end.

Brute force solution involves using a temporary data structure to store all non zero numbers after one iteration. Time complexity is $O(2n)$ and Space Complexity is $O(n)$.

Optimal Solution is using the two pointer approach. Our pointer i will simply be iterating through the array while j will always be at a zero. If element at i is non zero then we swap else we move on. Time Complexity is $O(n) + O(x)$ where x is occurrence of zero

Pseudocode :

```
moveZeros(arr, N) {
    j = -1
    for(int i = 0; i < N; i++) {
        if(arr[i] == 0) {
            j = i
            break
        }
    }
```

```
        i = j+1
        while(i < N && j < N) {
            if (arr[i] != 0) {
                swap(arr[i], arr[j]);
                j++
                i++
            } else i++
        }
}
```

Dry Run :