3. Given an array of integers. Write an algorithm that brings all nonzero elements to the left of the array, and returns the number of nonzero elements. The algorithm should operate in place, i.e. shouldn't create a new array. The order of the nonzero elements does not matter. The numbers that remain in the right portion of the array can be anything.

Example:

Example:
Given the array [1, 0, 2, 0, 0, 3, 4], A possible answer is [4, 1, 3, 2, ?, ?, ?],
4 non-zero elements, where "?" can be any number.
Code should have good complexity and minimize the number of writes to the array.
[1, 0, -2, 0, 0, 3, 4, 0, 0]
[4, 1, 3, -2, ?, ?, ?] return 4

```
public class Main {
    public static void main(String[] args) {
        int[] a = new int[] { 1, 0, 2, 0, 0, 3, 4 };
        int [] a1 = new int[] { 1, 0, -2, 0, 0, 3, 4, 0, 0};
        int cantNoNulos = ordNoNulos(a1);
        for (int elem : a1){
            System.out.println(elem);
        }
        System.out.println("Cantidad de no nulos: " + cantNoNulos);
    }

    public static int ordNoNulos(int[] arr) {
        /*
         aux se usará para saber en que posicion tengo que ir dejando los
elementos no nulos que encuentre
         Comienzo desde la primera posicion (0), y cada vez que encuentre un
elemento no nulo lo pongo
         en esa posicion y avanzo a la siguiente (aumento aux).
        */
        int aux = 0;
        for(int i = 0; i < arr.length; i++){
            if (arr[i] != 0){
                arr[aux] = arr[i];
                aux++;
            }
        }
        return aux++;
    }
}
```

Analysis

I can make an algorithm that goes through the whole array and every time it finds a non-null element it takes it to the beginning of the array.
For this, I can use an auxiliary variable that I will use as an index to move the non-null elements, starting at the beginning of the array (0), and every time I move an element I move the index to the next position.

Examples

```
a = [1, 0, 2, 0, 0, 3, 4]
/* Returns
1
2
3
4
0
3
4
Cantidad de no nulos: 4
*/
a1 = [1, 0, -2, 0, 0, 3, 4, 0, 0]
/* Returns
1
-2
3
4
0
3
4
0
0
Cantidad de no nulos: 4
*/
```

- What is the complexity of your algorithm?

My algorithm will carry non-null elements to the left with a single run of the array, so it will cycle n times, where n is the size of the array.