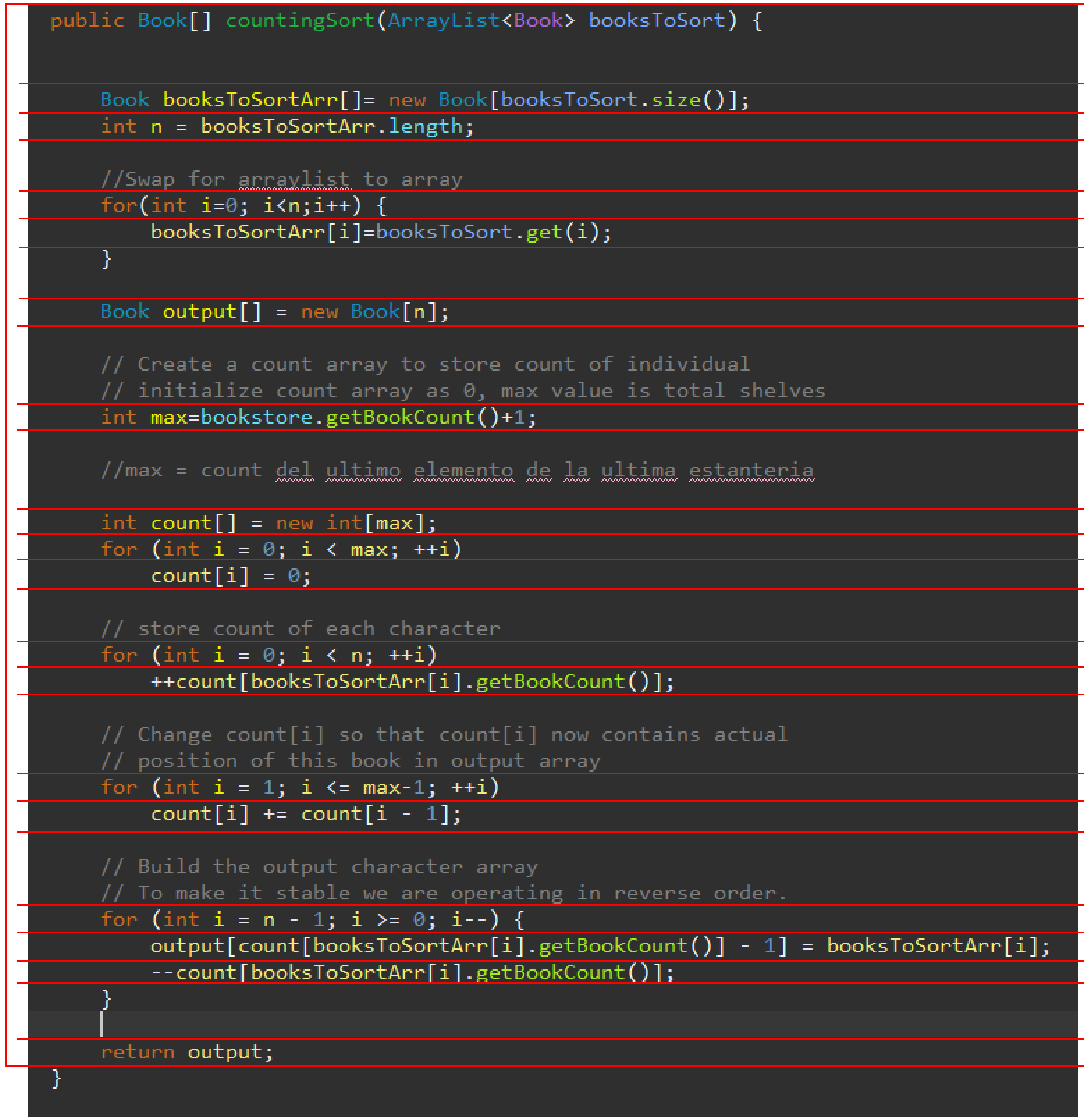
Temporal Espacial



```
public Book[] countingSort(ArrayList<Book> booksToSort) {

    Book booksToSortArr[]= new Book[booksToSort.size()];                                   1
    int n = booksToSortArr.length;                                                         1

    //Swap for arraylist to array
    for(int i=0; i<n;i++) {                                                                n+1
        booksToSortArr[i]=booksToSort.get(i);                                              n
    }

    Book output[] = new Book[n];                                                           1

    // Create a count array to store count of individual
    // initialize count array as 0, max value is total shelves
    int max=bookstore.getBookCount()+1;                                                    1

    //max = count del ultimo elemento de la ultima estanteria

    int count[] = new int[max];                                                            1
    for (int i = 0; i < max; ++i)                                                          n+1
        count[i] = 0;                                                                      n

    // store count of each character
    for (int i = 0; i < n; ++i)                                                            n+1
        ++count[booksToSortArr[i].getBookCount()];                                         n

    // Change count[i] so that count[i] now contains actual
    // position of this book in output array
    for (int i = 1; i <= max-1; ++i)                                                       n+1
        count[i] += count[i - 1];                                                          n

    // Build the output character array
    // To make it stable we are operating in reverse order.
    for (int i = n - 1; i >= 0; i--) {                                                     n+1
        output[count[booksToSortArr[i].getBookCount()] - 1] = booksToSortArr[i];           n
        --count[booksToSortArr[i].getBookCount()];                                         n
    }

    return output;                                                                         1
}
```

Complejidad espacial

Entrada array-> n
auxiliares max->1
output->n
count-> k
salida -> n

k= maximo valor del arreglo

O(k)

Temporal

Complejidad temporal total: 5(n+1)+6n+6
Complejidad temporal total:5n+5+6n+6
Complejidad temporal total:11n+11
Complejidad temporal total:11(n+1)

$$11(n+1) \longrightarrow O(n+k) \text{ en el peor de los casos}$$