

	Temporal
<pre>public ArrayList&lt;Book&gt; bubbleSort( ArrayList&lt;Book&gt; booksToSort) {</pre>	
<pre>    int n = booksToSort.size();</pre>	1
<pre>    for (int i = 0; i &lt; n-1; i++)</pre>	n+1
<pre>        for (int j = 0; j &lt; n-i-1; j++)</pre>	$\lfloor \frac{n(n+1)}{2} \rfloor + n$
<pre>            if (booksToSort.get(j).getShelve()==booksToSort.get(j+1).getShelve()){</pre>	$\lfloor \frac{n(n+1)}{2} \rfloor$
<pre>                if (booksToSort.get(j).getPosInShelve() &gt; booksToSort.get(j+1).getPosInShelve()) {</pre>	$\lfloor \frac{n(n+1)}{2} \rfloor$
<pre>                    Book temp = booksToSort.get(j);</pre>	$\lfloor \frac{n(n+1)}{2} \rfloor$
<pre>                    booksToSort.set(j, booksToSort.get(j+1));</pre>	$\lfloor \frac{n(n+1)}{2} \rfloor$
<pre>                    booksToSort.set(j+1,temp);</pre>	$\lfloor \frac{n(n+1)}{2} \rfloor$
<pre>                }</pre>	
<pre>            }else if (booksToSort.get(j).getShelve() &gt; booksToSort.get(j+1).getShelve()){</pre>	$\lfloor \frac{n(n+1)}{2} \rfloor$
<pre>                // swap</pre>	
<pre>                Book temp = booksToSort.get(j);</pre>	$\lfloor \frac{n(n+1)}{2} \rfloor$
<pre>                booksToSort.set(j, booksToSort.get(j+1));</pre>	$\lfloor \frac{n(n+1)}{2} \rfloor$
<pre>                booksToSort.set(j+1,temp);</pre>	$\lfloor \frac{n(n+1)}{2} \rfloor$
<pre>            }</pre>	
<pre>    return booksToSort;</pre>	1
<pre>}</pre>	

Complejidad espacial

Entrada n -> 1

auxiliares i ->1

j ->1

salida -> 1

0(1)

Complejidad temporal total: 2+(n+1)+([n(n+1/2)]+n)+9[n(n+1)/2]     $O(N^2)$  en el peor de los casos  
Complejidad temporal total:3+2n+10[n(n+1/2)]  
Complejidad temporal total:3+2n+5[n(n+1)]  
Complejidad temporal total:3+2n+5[n^2+n]  
Complejidad temporal total:3+2n+5n^2+5n  
Complejidad temporal total:3+7n+5n^2  
Complejidad temporal total:5n^2+7n+3