

Aclaraciones:

- Aunque existan decisiones que llevan a “caminos distintos” siempre se busca el peor caso. Por ello, algunas partes del código no serán incluidas en el cálculo de la complejidad.
- Solo se toma en cuenta la complejidad del método como tal y no se tienen en cuenta la complejidad de métodos como `whichShelf` puesto que su única función es permitir la comparación.

```
public void bubbleSortGames(String[] info){
```

```
String id = info[0];
```

```
String[] data = Arrays.copyOfRange(info, 1, info.length);
```

```
int n = data.length;
```

```
if(n==1){
```

```
    data[0]+=" "+whichShelf(data[0]);
```

```
}
```

```
else{
```

```
    for (int i = 0; i < n-1; i++) {
```

```
        for (int j = 0; j < n - i - 1; j++) {
```

```
            int a = whichShelf(data[j]);
```

```
            data[j]+=" "+a;
```

```
            int b = whichShelf(data[j+1]);
```

```
            data[j+1]+=" "+b;
```

```
            if (a >= b) {
```

```
                String temp = data[j];
```

```
                data[j] = data[j + 1];
```

```
                data[j + 1] = temp;
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

```
clients.add(new Client(id,data,counter));
```

```
counter++;
```

```
}
```

Suma de todas las líneas:

$$5 - 7n + 9 \left(\frac{n(n+1)}{2} \right) = \frac{9n^2}{2} + \frac{9n}{2} - 7n + 5 = \frac{9n^2}{2} - \frac{5n}{2}$$

Complejidad:

$$O(n^2)$$

```
public void insertionSortGames(String[] info){
```

```
String id = info[0]; 1
```

```
int i; 1
```

```
int j; 1
```

```
String aux=""; 1
```

```
String[] data = Arrays.copyOfRange(info, 1, info.length); 1
```

```
switch(data.length){ 1
```

```
case 1:
```

```
data[0]+=" "+whichShelf(data[0]);
```

```
break;
```

```
case 2:
```

```
int k=whichShelf(data[0]);
```

```
data[0]+=" "+k;
```

```
int q=whichShelf(data[1]);
```

```
data[1]+=" "+q;
```

```
if(q<k){
```

```
String aux2 = data[1];
```

```
data[1]=data[0];
```

```
data[0]=aux2;
```

```
}
```

```
break;
```

```
default:
```

```
for (i = 1; i < data.length; i++) { 1
```

```
aux = data[i]; 1-1
```

```
j = i - 1; 1-1
```

```
int a=whichShelf(data[j]); 1-1
```

```
data[j]+=" "+a; 1-1
```

```
int b=whichShelf(aux); 1-1
```

```
aux+=" "+b; 1-1
```

```
while ((j >= 0) && a >= b) {  $\frac{n(n+1)}{2} - 1$ 
```

```
data[j + 1] = data[j];  $\frac{n(n+1)}{2} - n$ 
```

```
j = j - 1;  $\frac{n(n+1)}{2} - n$ 
```

```
}
```

```
data[j + 1] = aux; 1-1
```

```
}
```

```
}
```

```
clients.add(new Client(id,data,counter)); 1
```

```
counter++; 1
```

```
}
```

Suma de las filas: $6n + 3\left(\frac{n(n+1)}{2}\right) = \frac{3n^2}{2} + \frac{3n}{2} + 6n = \frac{3n^2}{2} + \frac{15n}{2}$

Complejidad: $O(n^2)$