

COMPLEJIDAD

Algoritmo de ordenamiento

```
public static int[] insertionSort(int[] a) {  
  
    for (int i = 0; i < a.length; i++) { //c1+c2(n+1)+c3  
        int min = a[i]; //c4n  
        for (int j = i; j < a.length; j++) { //c5+c6(n(n+1))/2+c7  
            if (a[j] < min) { //c8*(n(n+1))/2  
                min = a[j]; //c9*(n(n+1))/2  
                int aux = a[i]; //c10*(n(n+1))/2  
                a[i] = min; //c11*(n(n+1))/2  
                a[j] = aux; //c12*(n(n+1))/2  
            }  
        }  
    }  
    return a; //c13  
}
```

$T(n) = c1 + c2(n+1) + c3 + c5 + c6 \cdot (n(n+1))/2 + c7 + c8 \cdot (n(n+1))/2 + c9 \cdot (n(n+1))/2 + c10 \cdot (n(n+1))/2 + c11 \cdot (n(n+1))/2 + c12 \cdot (n(n+1))/2 + c13$

$T(n)$ es $O(c6n^2 + n)$ Regla de la suma

$T(n)$ es $O(n^2)$ Regla del producto

Sumar arreglos

```
public static int suma(int[] a){  
    int suma = 0; // c_1  
    for(int i = 0; i < a.length; i++) // c2 + sum c3, i=0 to n  
        suma += a[i]; //sum c4, i=0 to n-1  
    return suma; //c5  
}
```

$$T(n) = c_1 + c_2 + c_3(n+1) + c_4n + c_5$$

$T(n)$ es $O(c_3n)$ Regla de la suma

$T(n)$ es $O(n)$ Regla del producto

En este ejercicio la complejidad con ciclos y la complejidad con recursión es la misma.