

# Análisis Insertion Sort

```
private static void insertaEnOrden(int[] A, int k) {  
    // HACER  
    int last = A[k];  
    int index = k - 1;  
    while((index ≥ 0) && A[index] > last) {  
        A[index + 1] = A[index];  
        index--;  
    }  
    A[index + 1] = last;  
}  
  
/*  
 * Ordena el array A[0...A.length-1] de menor a mayor valor, usando  
 * método insertaEnOrden.  
 */  
public static void OrdenPorInsercion(int[] A) {  
    // HACER  
    for(int j = 1; j < A.length; j++) {  
        insertaEnOrden(A, j);  
    }  
}
```

**A)** Este algoritmo se ejecutará K veces en el peor de los casos.

**B)** Haciendo uso del anterior algoritmo, el cual se ejecuta como máximo K veces (en este caso, K tomará el valor de J).

Como también sabemos que este algoritmo se ejecutará N-1 veces en el peor de los casos, pero que hará N comprobaciones en el peor de los casos...

Podemos analizarlo de la siguiente manera:

$$\begin{array}{l} j=1 \rightarrow 1 \\ j=2 \rightarrow 2 \\ \vdots \\ j=n \rightarrow n-1 \end{array} \Rightarrow \frac{n(n-1)}{2} = \frac{n^2}{2} - \frac{n}{2}$$

Por lo tanto podemos afirmar que...

$$O(n^2)$$