

# **Centro de Enseñanza Técnica Industrial**

[Estructura de datos y algoritmia]



# Métodos de Ordenamiento.

## Bubble Sort

Es un algoritmo simple de ordenamiento que funciona comparando e intercambiando repetidamente los elementos adyacentes en una lista hasta que la lista esté ordenada. El algoritmo recibe su nombre porque los elementos más grandes "burbujean" hacia el final de la lista, mientras que los más pequeños tienden a "flotar" hacia el principio.

El proceso se realiza iterativamente, comparando cada par de elementos adyacentes y realizando intercambios si es necesario. Este proceso se repite hasta que no se realizan más intercambios en una pasada completa, lo que indica que la lista está ordenada.

Aunque el bubble sort es fácil de entender y de implementar, no es eficiente para conjuntos de datos grandes, ya que su complejidad es cuadrática ( $O(n^2)$ ), lo que significa que su tiempo de ejecución crece rápidamente con el tamaño de la lista. Por esta razón, se prefiere en la práctica el uso de algoritmos de ordenamiento más eficientes para conjuntos de datos extensos.

## Selection Sort

Es otro algoritmo simple de ordenamiento. El proceso de ordenamiento se realiza dividiendo la lista en dos partes: una parte ordenada y otra desordenada. En cada iteración, el algoritmo busca el elemento más pequeño (o más grande, dependiendo de la variante) en la parte desordenada y lo intercambia con el primer elemento de la parte desordenada.

Este proceso se repite hasta que la parte desordenada se reduce a cero, lo que significa que la lista completa está ordenada. Aunque el selection sort es fácil de entender y su implementación es sencilla, su complejidad es cuadrática ( $O(n^2)$ ), lo que lo hace menos eficiente para conjuntos de datos grandes en comparación con algoritmos más avanzados.

A pesar de su falta de eficiencia para grandes conjuntos de datos, el selection sort puede ser útil en situaciones donde se prefiere la simplicidad y la memoria es limitada, ya que requiere un número constante de intercambios en cada iteración.

## Insertion Sort

Es otro algoritmo simple de ordenamiento. En este algoritmo, la lista se divide en dos partes: una parte ordenada y otra desordenada. En cada iteración, el algoritmo toma un elemento de la parte desordenada y lo inserta en la posición correcta dentro de la parte ordenada, desplazando los elementos mayores a la derecha.

El proceso se repite hasta que la parte desordenada se reduce a cero, lo que indica que la lista completa está ordenada. La clave del insertion sort es que, en cada paso, la parte ordenada sigue estando ordenada, y el algoritmo se construye gradualmente desde el principio de la lista hacia el final.

Aunque el insertion sort también tiene una complejidad cuadrática ( $O(n^2)$ ), puede ser más eficiente en la práctica que bubble sort y selection sort para conjuntos de datos pequeños o listas que ya están parcialmente ordenadas, ya que tiende a realizar menos comparaciones y desplazamientos. Sin embargo, para conjuntos de datos grandes, algoritmos más avanzados suelen ser preferidos.