



Hoja 9-2:

ANÁLISIS DE ALGORITMOS

Ejercicio 4:

Indica cuál es el número de asignaciones y de comparaciones de elementos que hay que realizar para insertar el elemento x en la tercera posición de la secuencia $(a, b, c, d, e, f, g, h, i)$, suponiendo las siguientes posibles representaciones para la secuencia:

- (i) En forma de *array*.
- (ii) Como una lista simplemente enlazada.

Da también el resultado para el caso hipotético en que la secuencia tuviese cien componentes.

Ejercicio 5:

Supongamos una lista enlazada de elementos. ¿Podemos utilizar en ella el método de ordenación por selección directa? En caso afirmativo escribe el algoritmo que realizaría la ordenación, en caso negativo realiza el algoritmo de ordenación por Inserción directa.

Ejercicio 6:

Indicar cuál será el número de comparaciones y asignaciones de elementos que habrá que realizar en un vector para introducir la secuencia $(10, 9, 8, 7, 6, 5, 4, 3, 2, 1)$, considerando los dos siguientes casos:

- (a) Los datos se añaden al final del vector.
- (b) Los datos se insertan de manera que el vector quede ordenado.

Ejercicio 7:

Las siguientes colecciones de 'n' elementos se pueden utilizar para almacenar datos. ¿Cuál es la complejidad computacional de la operación 'encontrar mínimo' en cada una de ellas?

- (a) Pila
- (b) Montículo de mínimos
- (c) Árbol binario de búsqueda
- (d) Lista ordenada con criterio ascendente
- (e) Lista ordenada con criterio descendente

Ejercicio 8:

El recorrido de un árbol binario de búsqueda en orden infijo nos proporciona la secuencia ordenada de sus elementos, con lo que podemos utilizar el árbol como una estructura auxiliar para ordenar un array de 'n' enteros. ¿Cuál sería el coste de esta ordenación?

Ejercicio 9:

Supongamos un vector con 100 elementos enteros que está ordenado. Si a ese vector le añadimos al final tres nuevos elementos, tal que el vector queda parcialmente desordenado, ¿cuál de los métodos de ordenación vistos en clase sería más eficiente para volver a ordenar el vector? ¿por qué?

Ejercicio 10:

¿Cuál sería el coste computacional de encontrar los elementos comprendidos entre unos ciertos valores 'a' y 'b' en cada una de las siguientes estructuras:

- (a) Pila
- (b) Cola
- (c) Árbol binario de búsqueda
- (d) Árbol binario de búsqueda hilvanado
- (e) Montículo de máximos