TEMA 8.- ALGORITMOS DE ORDENACION

E 1.- Escribir versiones recursivas de los algoritmos:

a) (CLASE) Inserción con centinela

b) Selección

c) Intercambio directo

Analizar los algoritmos obtenidos en asignaciones sobre el vector para los casos mejor y peor.

E 2.- (CLASE) Modificar el algoritmo de intercambio directo de forma que se determine el último lugar donde no se realizó intercambio y se eviten así iteraciones. Calcular los costes en función de la instrucción crítica (Sentencia SI).

E 3.- Realiza la traza del siguiente algoritmo para un vector: 53, 22, 41, 11, -3, 37, 0, 15, 38, 5

```
ALGORITMO SHELL
DATOS/RESULTADO: A:vector[0..n] de τ
METODO:
    incre ← ⌊n/2⌋
    MIENTRAS incre > 0 HACER
        PARA i ← incre+1 HASTA n HACER
            j ← i-incre
            MIENTRAS j > 0 HACER
                SI A[j] > A[j+incre] ENTONCES
                    A[j] ←→ A[j+incre]
                    j ← j-incre
                SINO j ← 0
                FSI
            FMIENTRAS
        FPARA
        incre ← ⌊incre/2⌋
    FMIENTRAS
FSHELL
```

E 4.- ¿Se puede escribir una versión del algoritmo de inserción directa que efectúe un número de asignaciones (sobre elementos del vector) que en el mejor caso sea menor que lineal? Si es posible escribirlo y recalcular los costes para el resto de casos (mejor, peor y medio). Sino, explicar el porqué.

E 5.- (CLASE) Escribir y analizar un algoritmo de ordenación basado en la siguiente idea: En una pasada se busca el mínimo y máximo de un vector y se colocan en la primera y última posición. A continuación el procedimiento continua de igual manera pero restringido al mismo vector sin la primera y última posición. Escribir una versión iterativa. Calcula los costes obtenidos para el caso mejor y peor en asignaciones y comparaciones sobre el vector.

E 6.- Escribir una versión del algoritmo de ordenación por mezcla que divida el vector en tres partes iguales y comparar sus costes con los costes de la versión estudiada en clase. ¿Qué coste asintótico tiene?.

E 7.- (CLASE) Realizar la traza del algoritmo Quicksort sobre el vector 5,3,2,3,1,2,3,7,2, tomando como pivote el elemento central en un caso y el primer elemento en el otro. ¿Cuál de las dos versiones es más eficiente en este caso particular?

E 8.- Escribir versiones iterativas de los algoritmos SUBIR y BAJAR sobre montículos de mínimos.

E 9.- (CLASE) Escribir un algoritmo que compruebe si un vector es un montículo de mínimos. Dar otra versión que de como resultado si el montículo es de máximos o de mínimos. Analizar los algoritmos

E 10.- Desarrollar un algoritmo de ordenación basado en la idea de un montículo ternario. Extender primeramente el concepto de montículo al caso de árboles ternarios y definir las operaciones básicas correspondientes.