

# Guía de ejercicios: Ordenamiento recursivo

## Consignas

### Ejercicio 1

Detallar los pasos del método *Shell Sort* para ordenar de menor a mayor los elementos del siguiente vector: 25 57 48 37 12 92 86 33

### Ejercicio 2

Detallar los pasos del método *Merge Sort* para ordenar de menor a mayor los elementos del siguiente vector: 1, 8, 6, 4, 10, 5, 3, 2, 22

### Ejercicio 3

Detallar los pasos del método *Quick Sort* para ordenar de menor a mayor los elementos del siguiente vector: 7 8 14 12 2 1 3 21

### Ejercicio 4

Determinar qué algoritmo de ordenación fue aplicado al vector:

3 , 8, 13, 25, 45, 23, 98, 58.

3 , 8, 13, 25, 45, 23, 98, 58.

3 , 8, 13, 25, 45, 23, 98, 58.

3 , 8, 13, 23, 25, 45, 58, 98.

### Ejercicio 5

Determinar qué algoritmo de ordenación fue aplicado al vector en las primeras tres pasadas:

5 2 7 3 1 8 2 6 9

2 3 1 2 5 7 8 6 9

1        2 3 2 5 6 7 8 9.

### Ejercicio 6

El siguiente vector contiene:

1 3 5 7 9 11 13 15 17 19 21 23

Utilizando el algoritmo de búsqueda binaria, trazar las etapas necesarias para encontrar el número 19.

#### Ejercicio 7

Desarrollar un algoritmo y luego codificarlo en un lenguaje de programación que:

- a) Genere 25 números aleatorios entre 100 y 300 y los almacene en un vector.
- b) Ordene el vector en forma creciente por el método de burbujas.
- c) Ordene el vector en forma creciente por el método *Shell*.
- d) Ordene el vector en forma decreciente por el método *Quick Sort*.
- e) Buscar si existe el número  $x$  introducido por teclado. Esta búsqueda deberá ser binaria.

#### Ejercicio 8

Se cargan dos vectores de  $n$  y  $m$  elementos; mostrarlos:

- a) Ordenar ambos vectores aplicando el método *Quick Sort*.
- b) Crear un vector mezclando los dos vectores.
- c) Ordenar este último vector por el método *Merge Sort*.