

GUIÓN DE LA PRÁCTICA 2

OBJETIVO:

- Algoritmos de ordenación y su estudio comparativo

1. TRES ALGORITMOS DE ORDENACIÓN “MALOS”

Se proporcionan ficheros Java para trabajar con los algoritmos siguientes vistos en teoría: *Inserción*, *Selección Directa* y *Burbuja*. **El alumno debe escribir el código específico de cada uno de los métodos.**

Son algoritmos *malos* porque son cuadráticos ($O(n^2)$) en sus casos mejor, peor y medio (excepto el de *Inserción* que en el caso mejor es lineal $O(n)$).

Para probar que los métodos funcionan bien se proporciona una clase *OrdenacionPruebas.java* → Tiene un argumento **n**, que es el tamaño del problema

Intenta comprender con detalle el funcionamiento de todos los algoritmos analizando los tiempos para diferentes tamaños del problema se proporciona una clase *OrdenacionTiempos* que deberías parametrizar para medir correctamente los tiempos respectivos en los distintos casos.

La forma de ejecutar los códigos podría ser la siguiente:

```
javac *.java
java alg77777777.p2.OrdenacionTiempos Tipo //Tipo es el tipo de
vector de número generado para ordenar
```

2. EL ALGORITMO DE ORDENACIÓN MEJOR: “QUICKSORT”

En este caso vas a estudiar el algoritmo de ordenación Rápido o Quicksort. Estúdialo en detalle, ya que es un algoritmo más elaborado que los otros. Completa el código cuando sea necesario y analiza los tiempos para diferentes tamaños del problema, concluyendo si los tiempos obtenidos son los esperados desde el punto de vista de la complejidad en cada caso.

- **RapidoCentral.java** → En este caso utilizamos como pivote el elemento central en cada iteración.
- **RapidoFatal.java** → Utiliza un pivote malo.
- **RapidoMedianaTres.java** → En este caso se elige la mediana a tres como pivote utilizando para ello el primer, el último y el elemento central en cada iteración. Además de completar el código, deberás medir los tiempos para este algoritmo utilizando la clase *OrdenacionTiempos*.

3. TRABAJO PEDIDO

Se le pide programar lo necesario para rellenar 4 tablas de tiempos (para *Inserción*, *Selección*, *Burbuja* y *Quicksort* con pivote utilizando mediana a tres). Cada tabla será así:

Tiempos MÉTODO x:

<i>N</i>	<i>t ordenado</i>	<i>t inverso</i>	<i>t aleatorio</i>
10000
20000
40000
80000
160000
320000
640000
1280000

Hasta “heap overflow” o casque			

Las clases que programe las incluirá dentro del paquete alg<dnipropio>.p2. Si utiliza Eclipse llamar al proyecto prac02_Ordena<UOpropio>

Se entregará, por separado, un documento con las 4 tablas cubiertas, y los ficheros fuente de las clases que haya tenido que programar o modificar junto con una explicación coherente de los resultados.

Esto se realizará a través de la tarea que se habilitará en el campus virtual. El plazo límite es un día antes de la próxima sesión de prácticas.