

## **Práctica 5: Implementación de métodos de ordenación**

---

### **Objetivo**

En esta práctica se trabaja los algoritmos de ordenación interna, tanto la implementación en lenguaje C++ como el estudio de la complejidad computacional de dichos algoritmos.

### **Entrega**

Esta práctica se realizará en dos sesiones de laboratorio en las siguientes fechas:

Sesión tutorada: 10, 11 y 12 de abril de 2018;

Sesión de entrega: 17, 18 y 19 de abril de 2018.

Durante las sesiones de laboratorio se podrán proponer modificaciones y mejoras en el enunciado de la práctica.

### **Enunciado**

Implementar en lenguaje C++ los siguientes métodos de ordenación [1][2]:

- Inserción
- Por Intercambio: Método de la Burbuja
- HeapSort
- QuickSort
- Por Incrementos Decrecientes (ShellSort): debe permitir seleccionar la constante de reducción alfa, siendo  $0 < \alpha < 1$

Realizar un programa en C++ para observar la traza del funcionamiento de cada algoritmo para secuencias pequeñas.

Realizar un estudio empírico del rendimiento de los métodos de ordenación al incrementar el tamaño del array a ordenar.

El estudio del rendimiento requiere desarrollar un programa en C++ que cuente el número de operaciones de comparación de clave que se realizan durante la ordenación de un array. El programa utilizará un contador de comparaciones, que se inicializa a cero antes de la ejecución de cada método de ordenación y se incrementa con cada ejecución de una operación de comparación de claves. Al finalizar la ordenación el valor contenido en el contador se utiliza para actualizar una estadística que registra los valores mínimo, máximo y media del número de comparaciones obtenidos. Para que estos valores estadísticos sean significativos el experimento de ordenación debe repetirse un número suficiente de veces. En la sección de notas de implementación se indica el formato de visualización de los datos obtenidos en la ejecución del programa.

## Notas de implementación

La implementación de cada método de ordenación se realizará mediante una plantilla de función en la que se especificará el tipo de elementos a ordenar, y la función recibirá como parámetros la secuencia a ordenar de tipo `Clave` y su tamaño:

```
nombre_método<Clave> (vector<Clave>, tamaño)
```

\*\*\*

Para probar el funcionamiento de los métodos de ordenación y realizar el estudio de su comportamiento se utilizarán valores de clave del tipo `DNI` (clase definida en el enunciado de la práctica 4). Las secuencias a ordenar se generarán de manera aleatoria.

\*\*\*

Se deben realizar dos programas:

1. **Programa Modo Demostración:** Se utilizan secuencias pequeñas (máximo de 25 elementos) para probar el funcionamiento de un algoritmo determinado. El programa realizará la siguiente secuencia de pasos:
  1. Solicitar los parámetros necesarios para su ejecución:
    - a. N: Tamaño de la secuencia a ordenar.
    - b. Algoritmo a ejecutar.
  2. Generar de forma aleatoria una secuencia de tamaño N de objetos `DNI`.
  3. Ejecución del algoritmo seleccionado: al pulsar una tecla, se mostrará el resultado de cada comparación, destacando los elementos del vector que son comparados y cómo queda la secuencia tras la comparación y la acción realizada.
2. **Programa Modo Estadística:** El programa realizará la siguiente secuencia de pasos:
  1. Solicitar los parámetros del experimento:
    - a. N: Tamaño de la secuencia a ordenar.
    - b. Número de pruebas, `nPruebas`: Número de veces que se repite la ejecución de cada método.
  2. Para cada uno de los algoritmos se crea un banco de prueba con `nPruebas` secuencias de N valores de tipo `DNI` generados de forma aleatoria.
  3. El experimento para estudiar el comportamiento consiste en, para cada método de ordenación:

- a. Inicializar a cero los contadores de comparaciones de claves. Valores mínimo, acumulado y máximo.
- b. Realizar la ordenación de las `nPruebas` secuencias de `N` claves del banco de prueba. Para cada ejecución se cuenta el número de comparaciones realizadas, y se actualizan los valores mínimo, máximo y acumulado.
- c. Al finalizar el experimento se presentan los valores mínimo, máximo y medio del número de comparaciones de claves contabilizados.

A continuación se muestra el formato de salida con los resultados de la ejecución:

	Número de Comparaciones		
	Mínimo	Medio	Máximo
Método 1	xxxx	xxxx	xxxx
Método 2	xxxx	xxxx	xxxx
...			

\*\*\*

De forma opcional, se puede utilizar el Programa Modo Estadística desarrollado para realizar un estudio de la variación del comportamiento en los métodos de ordenación cuando se incrementa el tamaño de la secuencia a ordenar.

Además, también se pueden implementar el resto de algoritmos de ordenación vistos en clase.

## Referencias

- [1] Apuntes de clase.  
[2] [https://es.wikipedia.org/wiki/Algoritmo\\_de\\_ordenamiento](https://es.wikipedia.org/wiki/Algoritmo_de_ordenamiento)