

Práctica 5: Implementación de métodos de ordenación

1. Objetivo

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

2. Entrega

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

Sesión tutorada online: 21 de abril de 2020, a las 8:30 y a las 14:40;

Sesión entrega online: 30 de abril, a las 8:30 y a las 14:30

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

3. Enunciado

Implementar en lenguaje C++, al menos, 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 $0 < \text{alfa} < 1$

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

Realizar un estudio empírico del rendimiento de los métodos de ordenación.

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.

El estudio del rendimiento de los métodos de ordenación consiste en realizar distintas ejecuciones de cada método implementado variando el tamaño del array a ordenar y anotar los valores mínimo, media y máximo del número de comparaciones obtenidos en cada ejecución.

4. 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 (`Clave`). La función recibirá como parámetros la secuencia a ordenar con elementos 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.

Para su uso en esta práctica será necesario implementar en la clase `DNI` todos los operadores de comparación utilizados en los métodos de ordenación.

Se deben realizar dos programas principales:

A. 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.

B. 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
...			

Además:

Se puede utilizar el Programa Modo Estadística desarrollado para realizar el estudio de la variación del comportamiento en los métodos de ordenación cuando se ejecuta varias veces, incrementando el tamaño de la secuencia a ordenar.

También se puede implementar el resto de algoritmos de ordenación vistos en clase.

5. Referencias

- [1] Apuntes de clase.
- [2] https://es.wikipedia.org/wiki/Algoritmo_de_ordenamiento