UNIDAD TEMÁTICA 9 – CLASIFICACIÓN

Trabajo de Aplicación 1

EJERCICIO 1

1. Desarrolla en seudocódigo el método de ordenación por Inserción Directa, y analiza detalladamente el orden del tiempo de ejecución.
2. Utilizando el método de Inserción directa sobre el siguiente conjunto de datos, ilustra en cada paso cómo se van clasificando. Registra, en cada iteración, la cantidad de **comparaciones *C*** y **movimientos *M*** realizados

   256 - 458 - 655 - 298 - 043 - 648 - 778 - 621 - 655 - 019 - 124 - 847

3. Responde las preguntas presentadas en pantalla

Trabajo de Aplicación 1

EJERCICIO 2

1. Utilizando el método de Shell con la secuencia de incrementos ***(5, 3, 1)*** sobre el siguiente conjunto de datos, ilustra en cada paso cómo se van clasificando. Registra, en cada iteración, la cantidad de **comparaciones *C*** y **movimientos *M*** realizados

   256 - 458 - 655 - 298 - 043 - 648 - 778 - 621 - 655 - 019 - 124 - 847

2. Analiza detalladamente el orden del tiempo de ejecución del método de ordenación "SHELLSORT".
3. Responde las preguntas presentadas en pantalla

Trabajo de Aplicación 1

EJERCICIO 3

1. Dado el siguiente vector de claves, ordénalo paso a paso usando el método de BURBUJA. Registra, en cada iteración, la cantidad de **comparaciones *C*** y **movimientos *M*** realizados

256 - 458 - 655 - 298 - 043 - 648 - 778 - 621 - 655 - 019 - 124 - 847

2. Analiza el orden del tiempo de ejecución del método de ordenación “BURBUJA”.
3. Indica una forma en que pueda reducirse el tiempo de ejecución
4. Responde las preguntas presentadas en pantalla

Trabajo de Aplicación 1

EJERCICIO 4

Utilizando las clases JAVA provistas por la Cátedra "**TClasificador.java**" y **"GeneradorDatosGenerico.java"**, genera un paquete **"UT9"** para contener los programas de prueba de los algoritmos de clasificación.

Se requiere:

PARTE 1:

Estudiar el funcionamiento de la clase "***GeneradorDatosGenerico.java***" y responder las preguntas presentadas en pantalla.

1) ¿Cómo se genera un vector monótonamente ascendente
2) ¿Cómo se genera un vector monótonamente descendente?
3) ¿Cómo se genera un vector con valores aleatorios? ¿pueden existir claves repetidas? ¿cuál es el orden del tiempo de ejecución de este método?
4) ¿Cuántos elementos contiene el vector de datos generado? ¿cómo se puede modificar esta clase para que la cantidad de elementos del vector sea parametrizable?
5) ¿Cómo podemos verificar que un conjunto está ordenado? ¿cuál sería el orden del tiempo de ejecución de un algoritmo que lo haga?

PARTE 2:

La clase ***"TClasificador.java***" contiene ya métodos para los algoritmos de **inserción directa**, **shellsort** y **burbuja**. Estos métodos **contienen errores.**

1) Declarar un vector estático en el "*main*" con unas pocas (no más de 5) claves enteras, desordenadas
2) Probar la ejecución del método "***clasificar***" del *TClasificador*, para invocar el método Inserción Directa.
3) Observar el resultado emitido por consola.
4) Encontrar y reparar los eventuales errores en el método "***OrdenarPorInsercion***"
5) Volver a ejecutar, y verificar el orden de la salida mediante la ejecución de un método "***estaOrdenado***" que tome como parámetro el vector resultante de la ordenación.
6) Probar la ejecución con vectores con datos ascendentes, descendentes y aleatorios. Desarrollar casos de test para verificar el correcto funcionamiento.
7) ¿cuál es el tiempo de ejecución para cada tipo de vector (tamaño y orden)?
8) Repetir los pasos 1 a 7 para los métodos ***Shellsort*** y ***Burbuja***.

NOTA: SE SUGIERE UTILIZAR CONJUNTOS DE DATOS PEQUEÑOS PARA REALIZAR LAS PRIMERAS PRUEBAS.