

**LABORATORIO 3: METODOS DE ORDENAMIENTO LINEAL EN  
ESTRUCTURAS DE DATOS ESTATICAS**

**CONTENIDOS**

1	<a href="#">Objetivo</a>	<a href="#">2</a>
2	<a href="#">Conceptualizacion</a>	<a href="#">2</a>
3	<a href="#">Tad busqueda</a>	<a href="#">4</a>
4	<a href="#">Aplicacion</a>	<a href="#">7</a>
5	<a href="#">Consulta</a>	<a href="#">7</a>

**OBJETIVOS**

- Conceptuar el proceso de ordenamiento dentro de Estructuras de Datos
- Identificar de forma practica el comportamiento del Metodo de Ordenamiento Burbuja
- Identificar de forma practica el comportamiento del Metodo de Ordenamiento por Insercion
- Identificar de forma practica el comportamiento del Metodo de Ordenamiento por Seleccion
- Identificar de forma practica el comportamiento del Metodo de Ordenamiento Shell
- Identificar de forma practica el comportamiento del Metodo de Ordenamiento Quicksort
- Identificar de forma practica el comportamiento del Metodo de Ordenamiento Mergesort

**CONCEPTUALIZACION**

El **Ordenamiento** es el proceso de organizacion de datos de manera creciente o decreciente respetando las caracter sticas de un campo espec fico de los registros propios de la estructura de datos.

***METODOS DE ORDENAMIENTO***

Dada la tecnica empleada para alcanzar el ordenamiento de los registros –los metodos seran tratados para la clasificacion en orden ascendente-, se cuenta con los siguientes metodos:

Sea la estructura de enteros tipo vector:

8	5	4	9	7	2	1	2	9	6	3	0	7	8	5
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

***Ordenamiento Burbuja. (Hundimiento)***

Metodo fundamentado en la realizacion de multiples pasadas a la estructura, de manera que en cada pasada se garantice que el elemento mas grande de la subestructura se “hunda” al termino.

En la primera pasada se obtendra:

5	4	8	7	2	1	2	9	6	3	0	7	8	5	9
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Notese que el valor 9 se fue hundiendo en cada comparacion hasta llegar al fondo, lo que garantiza que en pasadas posteriores ya no se haga necesario comparar con esta posicion. Segunda pasada:

4	5	7	2	1	2	8	6	3	0	7	8	5	9	9
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Tercera pasada:

4	5	2	1	2	7	6	3	0	7	8	5	8	9	9
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Se hacen  $N - 1$  pasadas para obtener:

0	1	2	2	3	4	5	5	6	7	7	8	8	9	9
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

**Ordenamiento por Insercion. (Baraja)**

Se considera que una parte de la estructura ya se encuentra clasificada (ordenada), as el registro del ndice 1 se ordenara dentro de la parte ordenada de la subestructura que contiene el elemento de la posicion 0, de modo que se tiene:

5	8													
---	---	--	--	--	--	--	--	--	--	--	--	--	--	--

Notese que se ha producido un corrimiento de posiciones. Ahora la subestructura ordenada cuenta con dos elementos, por tanto, se pretende ubicar el elemento de la posicion 2, produciendose:

4	5	8												
---	---	---	--	--	--	--	--	--	--	--	--	--	--	--

De tal manera que al aplicar el metodo en forma repetida se obtendra:

0	1	2	2	3	4	5	5	6	7	7	8	8	9	9
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

**Ordenamiento por Seleccion.**

Metodo basado en la busqueda y ubicacion del menor de los elementos de la estructura en la primera posicion, luego se ubica el segundo menor y se posiciona en la respectiva posicion y as sucesivamente, obteniendose:

8	5	4	9	7	2	1	2	9	6	3	0	7	8	5
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

0	5	4	9	7	2	1	2	9	6	3	8	7	8	5
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Notese que se ha producido el intercambio de posiciones.

0	1	4	9	7	2	5	2	9	6	3	8	7	8	5
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Se repite el metodo hasta alcanzar al termino la organizacion pretendida.

**Ordenamiento Tipo Shell.**

Se parte del establecimiento de un numero de desplazamiento –inferior al tamaño de la estructura-, que determinara los elementos a comparar, produciendo cierto numero de intercambios por cada recorrido, luego se reduce el numero de desplazamiento hasta alcanzar posiblemente un ordenamiento tipo burbuja.

Salto: 3 (No es riguroso el inicio con este valor)

8	5	4	9	7	2	1	2	9	6	3	0	7	8	5
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Con este salto se comparan los elementos de los índices (0,3), (1,4), (2,5),... (11,14), obteniéndose al termino del primer recorrido:

8	5	2	1	2	4	6	3	0	7	7	5	9	8	9
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Se inicia el segundo recorrido, dado que se produjeron cambios, con el mismo valor de desplazamiento:

1	2	2	6	3	0	7	5	4	8	7	5	9	8	9
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Se inicia el tercer recorrido con el mismo valor de desplazamiento, dado que se produjeron cambios:

1	2	0	6	3	2	7	5	4	8	7	5	9	8	9
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Se inicia el cuarto recorrido con el mismo valor de desplazamiento, dado que se produjeron cambios:

1	2	0	6	3	2	7	5	4	8	7	5	9	8	9
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Dado que no hubo cambios, el valor de desplazamiento se reduce, generalmente con una funcion de reduccion, supongamos que se ha reducido a un valor de 1:

1	0	2	3	2	6	5	4	7	7	5	8	8	9	9
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Y se repite con el mismo valor hasta alcanzar al termino el orden deseado.

**Ordenamiento Quicksort.**

Fundamentado en el hecho que es mas rapido y facil ordenar dos estructuras pequenas que una grande. Parte del hecho de seleccionar un pivote y en la sublista de izquierda ubica uno mayor y lo intercambia con uno menor de la derecha, repitiendo el proceso hasta que no exista elemento para con el cual producir intercambio, situacion cuando se cambia de pivote y se procede a ordenar las dos subestructuras.

>							P							<															
<table border="1" style="width: 100%; text-align: center; border-collapse: collapse;"> <tr> <td>8</td><td>5</td><td>4</td><td>9</td><td>7</td><td>2</td><td>1</td><td>2</td><td>9</td><td>6</td><td>3</td><td>0</td><td>7</td><td>8</td><td>5</td> </tr> </table>															8	5	4	9	7	2	1	2	9	6	3	0	7	8	5
8	5	4	9	7	2	1	2	9	6	3	0	7	8	5															

En la parte baja se selecciona un elemento mayor al pivote y en la parte derecha su complementario, para intercambiarse:

>							<	P																					
<table border="1" style="width: 100%; text-align: center; border-collapse: collapse;"> <tr> <td>0</td><td>5</td><td>4</td><td>9</td><td>7</td><td>2</td><td>1</td><td>2</td><td>9</td><td>6</td><td>3</td><td>8</td><td>7</td><td>8</td><td>5</td> </tr> </table>															0	5	4	9	7	2	1	2	9	6	3	8	7	8	5
0	5	4	9	7	2	1	2	9	6	3	8	7	8	5															

<	>							P																					
<table border="1" style="width: 100%; text-align: center; border-collapse: collapse;"> <tr> <td>0</td><td>1</td><td>4</td><td>9</td><td>7</td><td>2</td><td>5</td><td>2</td><td>9</td><td>6</td><td>3</td><td>8</td><td>7</td><td>8</td><td>5</td> </tr> </table>															0	1	4	9	7	2	5	2	9	6	3	8	7	8	5
0	1	4	9	7	2	5	2	9	6	3	8	7	8	5															

Dado que no se hizo posible encontrar un elemento menor a pivote en la parte derecha del seleccionado como mayor a pivote, se intercambian el mayor y el pivote:

							P																						
<table border="1" style="width: 100%; text-align: center; border-collapse: collapse;"> <tr> <td>0</td><td>1</td><td>2</td><td>9</td><td>7</td><td>2</td><td>5</td><td>4</td><td>9</td><td>6</td><td>3</td><td>8</td><td>7</td><td>8</td><td>5</td> </tr> </table>															0	1	2	9	7	2	5	4	9	6	3	8	7	8	5
0	1	2	9	7	2	5	4	9	6	3	8	7	8	5															

Se reinicia el proceso de busqueda:

>							P							<															
<table border="1" style="width: 100%; text-align: center; border-collapse: collapse;"> <tr> <td>0</td><td>1</td><td>2</td><td>9</td><td>7</td><td>2</td><td>5</td><td>4</td><td>9</td><td>6</td><td>3</td><td>8</td><td>7</td><td>8</td><td>5</td> </tr> </table>															0	1	2	9	7	2	5	4	9	6	3	8	7	8	5
0	1	2	9	7	2	5	4	9	6	3	8	7	8	5															

>	<						P																						
<table border="1" style="width: 100%; text-align: center; border-collapse: collapse;"> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>7</td><td>2</td><td>5</td><td>4</td><td>9</td><td>6</td><td>9</td><td>8</td><td>7</td><td>8</td><td>5</td> </tr> </table>															0	1	2	3	7	2	5	4	9	6	9	8	7	8	5
0	1	2	3	7	2	5	4	9	6	9	8	7	8	5															

>	<						P																						
<table border="1" style="width: 100%; text-align: center; border-collapse: collapse;"> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>2</td><td>7</td><td>5</td><td>4</td><td>9</td><td>6</td><td>9</td><td>8</td><td>7</td><td>8</td><td>5</td> </tr> </table>															0	1	2	3	2	7	5	4	9	6	9	8	7	8	5
0	1	2	3	2	7	5	4	9	6	9	8	7	8	5															

<	>						P																						
<table border="1" style="width: 100%; text-align: center; border-collapse: collapse;"> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>2</td><td>7</td><td>5</td><td>4</td><td>9</td><td>6</td><td>9</td><td>8</td><td>7</td><td>8</td><td>5</td> </tr> </table>															0	1	2	3	2	7	5	4	9	6	9	8	7	8	5
0	1	2	3	2	7	5	4	9	6	9	8	7	8	5															





**Postcondicion:** Todos los campos de cada registro corresponden a las reglas de validacion especificadas.

***LlenaDatosAutomatico.***

**Descripcion:** El procesador brinda aleatoriamente la informacion de cada uno de los registros de la estructura.

**Datos de ingreso:** Vector Nomina de tipo Estructura Persona.

**Datos de salida:** Vector Nomina modificado.

**Precondicion:** True

**Postcondicion:** Todos los campos de cada registro corresponden a las reglas de validacion especificadas.

**OPERACIONES**

***OrdenamientoBurbuja.***

**Descripcion:** Esta operacion permite ordenar la respectiva estructura, empleando el metodo burbuja, segun el valor del campo Identificacion.

**Datos de ingreso:** Vector Nomina de tipo Estructura Persona.

**Datos de salida:** Vector Nomina modificado.

**Precondicion:** Cada una de los 80 campos del vector debe poseer la informacion validada.

**Postcondicion:** La ubicacion de cada uno de los 80 campos de la estructura estara clasificada de manera ascendente segun el valor del campo clave Identificacion.

***OrdenamientoInsercion.***

**Descripcion:** Esta operacion permite ordenar la respectiva estructura, empleando el metodo Insercion, segun el valor del campo Identificacion.

**Datos de ingreso:** Vector Nomina de tipo Estructura Persona.

**Datos de salida:** Vector Nomina modificado.

**Precondicion:** Cada una de los 80 campos del vector debe poseer la informacion validada.

**Postcondicion:** La ubicacion de cada uno de los 80 campos de la estructura estara clasificada de manera ascendente segun el valor del campo clave Identificacion.

***OrdenamientoSeleccion.***

**Descripcion:** Esta operacion permite ordenar la respectiva estructura, empleando el metodo Seleccion, segun el valor del campo Identificacion.

**Datos de ingreso:** Vector Nomina de tipo Estructura Persona.

**Datos de salida:** Vector Nomina modificado.

**Precondicion:** Cada una de los 80 campos del vector debe poseer la informacion validada.

**Postcondicion:** La ubicacion de cada uno de los 80 campos de la estructura estara clasificada de manera ascendente segun el valor del campo clave Identificacion.

***OrdenamientoShell.***

**Descripcion:** Esta operacion permite ordenar la respectiva estructura, empleando el metodo Shell, segun el valor del campo Identificacion.

**Datos de ingreso:** Vector Nomina de tipo Estructura Persona.

**Datos de salida:** Vector Nomina modificado.

**Precondicion:** Cada una de los 80 campos del vector debe poseer la informacion validada.

**Postcondicion:** La ubicacion de cada uno de los 80 campos de la estructura estara clasificada de manera ascendente segun el valor del campo clave Identificacion.

#### ***OrdenamientoQuickSort.***

**Descripcion:** Esta operacion permite ordenar la respectiva estructura, empleando el metodo quickSort, segun el valor del campo Identificacion.

**Datos de ingreso:** Vector Nomina de tipo Estructura Persona.

**Datos de salida:** Vector Nomina modificado.

**Precondicion:** Cada una de los 80 campos del vector debe poseer la informacion validada.

**Postcondicion:** La ubicacion de cada uno de los 80 campos de la estructura estara clasificada de manera ascendente segun el valor del campo clave Identificacion.

#### ***OrdenamientoIntercalacion.***

**Descripcion:** Esta operacion permite ordenar la respectiva estructura, empleando el metodo Intercalacion, segun el valor del campo Identificacion.

**Datos de ingreso:** Vector Nomina de tipo Estructura Persona.

**Datos de salida:** Vector Nomina modificado.

**Precondicion:** Cada una de los 80 campos del vector debe poseer la informacion validada.

**Postcondicion:** La ubicacion de cada uno de los 80 campos de la estructura estara clasificada de manera ascendente segun el valor del campo clave Identificacion.

### **APLICACION**

1. Implemente el TAD mencionado preliminarmente
2. Modifique por lo menos dos de las operaciones implementadas a partir del TAD, de manera que operen de forma recursiva, guardando sus resultados en una carpeta alterna.

### **CONSULTAS**

- ¿Que es una Estructura Dinamica de Datos?
- ¿Cual es la diferencia entre EDD y Estructura Estatica de Datos?
- ¿Por que un vector predefinido en el programa se considera estatico?
- ¿Como se solicita memoria al procesador?
- ¿Que es una Lista?