



TUDIMUM
FORMACIÓN

GRUPO

Taller Tema 3: Búsqueda y Ordenación de tablas

Resueltos en JAVA



Búsqueda de un elemento en una tabla no ordenada

```
package es.studium.Buscar;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Buscar
{
    public static void main(String args[]) throws IOException
    {
        int tabla[] = new int[10];
        int i, elemento;
        BufferedReader lectura = new BufferedReader(new
        InputStreamReader(System.in));
        boolean encontrado = false;
        for(i=0;i<10;i++)
        {
            System.out.println("Dame un número:");
            tabla[i]= Integer.parseInt(lectura.readLine());
        }
        System.out.println("Dame un número a buscar:");
        elemento= Integer.parseInt(lectura.readLine());
        i=0;
        while((!encontrado)&&(i<10))
        {
            if(tabla[i]==elemento)
            {
                encontrado = true;
            }
            i++;
        }
        if(encontrado)
        {
            System.out.println("El número buscado está en la
            posición "+i);
        }
        else
        {
            System.out.println("El número buscado no se
            encuentra");
        }
    }
}
```

Ordenación de una tabla

Intercambio:

```
package es.studium.Intercambio;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Intercambio
{
    public static void main(String[] args) throws IOException
    {
        BufferedReader lectura = new BufferedReader(new
        InputStreamReader(System.in));
        int array[] = new int[10];
        int i, j, aux;
        // Rellenar la tabla
        for(i=0;i<10;i++)
        {
            System.out.println("Dame el valor de la tabla
["+i+"]");
            array[i] = Integer.parseInt(lectura.readLine());
        }
        // Ordenar la tabla
        for(i=0;i<9;i++)
        {
            for(j=i+1;j<10;j++)
            {
                if(array[i] > array[j])
                {
                    aux = array[i];
                    array[i] = array[j];
                    array[j] = aux;
                }
            }
        }
        // Mostrar los resultados
        System.out.println("La tabla ordenada sería:");
        for(i=0;i<10;i++)
        {
            System.out.print(array[i]+" ");
        }
    }
}
```

Selección:

```

package es.studium.Seleccion;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Seleccion
{
    public static void main(String[] args) throws IOException
    {
        BufferedReader lectura = new BufferedReader(new
        InputStreamReader(System.in));
        int tabla[] = new int[10];
        int i, j, indice_menor, aux;
        // Rellenar la tabla
        for(i=0;i<10;i++)
        {
            System.out.println("Dame el valor de la tabla
["+i+"]");
            tabla[i] = Integer.parseInt(lectura.readLine());
        }
        // Ordenar la tabla
        for(i=0;i<9;i++)
        {
            indice_menor = i;
            for(j=i+1;j<10;j++)
            {
                if(tabla[j] < tabla[indice_menor])
                {
                    indice_menor = j;
                }
            }
            aux = tabla[i];
            tabla[i] = tabla[indice_menor];
            tabla[indice_menor] = aux;
        }
        // Mostrar los resultados
        System.out.println("La tabla ordenada sería:");
        for(i=0;i<10;i++)
        {
            System.out.print(tabla[i]+" ");
        }
    }
}

```

Inserción:

```
package es.studium.Insercion;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Insercion
{
    public static void main(String[] args) throws IOException
    {
        BufferedReader lectura = new BufferedReader(new
        InputStreamReader(System.in));
        int tabla[] = new int[10];
        int i, j, aux;
        // Rellenar la tabla
        for(i=0;i<10;i++)
        {
            System.out.println("Dame el valor de la tabla
["+i+"]");
            tabla[i] = Integer.parseInt(lectura.readLine());
        }
        // Ordenar la tabla
        for(i=0;i<10;i++)
        {
            j = i;
            aux = tabla[i];
            while ((j>0)&&(aux < tabla[j-1]))
            {
                tabla[j] = tabla[j-1];
                j--;
            }
            tabla[j] = aux;
        }
        // Mostrar los resultados
        System.out.println("La tabla ordenada sería:");
        for(i=0;i<10;i++)
        {
            System.out.print(tabla[i]+" ");
        }
    }
}
```

Burbuja:

```

package es.studium.Burbuja;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Burbuja
{
    public static void main(String[] args) throws IOException
    {
        BufferedReader lectura = new BufferedReader(new
        InputStreamReader(System.in));
        int tabla[] = new int[10];
        int i, j, aux;
        // Rellenar la tabla
        for(i=0;i<10;i++)
        {
            System.out.println("Dame el valor de la tabla
["+i+"]");
            tabla[i] = Integer.parseInt(lectura.readLine());
        }
        // Ordenar la tabla
        for(i=0;i<9;i++)
        {
            for(j=0;j<9;j++)
            {
                if(tabla[j] > tabla[j+1])
                {
                    aux = tabla[j];
                    tabla[j] = tabla[j+1];
                    tabla[j+1] = aux;
                }
            }
        }
        // Mostrar los resultados
        System.out.println("La tabla ordenada sería:");
        for(i=0;i<10;i++)
        {
            System.out.print(tabla[i]+" ");
        }
    }
}

```

Hundimiento:

```

package es.studium.Hundimiento;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Hundimiento
{
    public static void main(String[] args) throws IOException
    {
        BufferedReader lectura = new BufferedReader(new
        InputStreamReader(System.in));
        int tabla[] = new int[10];
        int i, j, aux;
        // Rellenar la tabla
        for(i=0;i<10;i++)
        {
            System.out.println("Dame el valor de la tabla
["+i+"]");
            tabla[i] = Integer.parseInt(lectura.readLine());
        }
        // Ordenar la tabla
        for(i=1;i<9;i++)
        {
            for(j=9;j>=i;j--)
            {
                if(tabla[j-1] > tabla[j])
                {
                    aux = tabla[j];
                    tabla[j] = tabla[j-1];
                    tabla[j-1] = aux;
                }
            }
        }
        // Mostrar los resultados
        System.out.println("La tabla ordenada sería:");
        for(i=0;i<10;i++)
        {
            System.out.print(tabla[i]+" ");
        }
    }
}

```

Shell:

```

package es.studium.Shell;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Shell
{
    public static void main(String[] args) throws IOException
    {
        BufferedReader lectura = new BufferedReader(new
        InputStreamReader(System.in));
        int array[] = new int[10];
        int salto, aux, i;
        boolean cambios;
        // Rellenar la tabla
        for(i=0;i<10;i++)
        {
            System.out.println("Dame el valor de la tabla
["+i+"]");
            array[i] = Integer.parseInt(lectura.readLine());
        }
        // Ordenar la tabla
        for(salto=5;salto!=0;salto=salto/2)
        {
            cambios = true;
            while(cambios==true)
            {
                cambios = false;
                for(i=salto;i<10;i++)
                {
                    if(array[i-salto] > array[i])
                    {
                        aux = array[i];
                        array[i] = array[i-salto];
                        array[i-salto] = aux;
                        cambios = true;
                    }
                }
            }
        }
        // Mostrar los resultados
        System.out.println("La tabla ordenada sería:");
        for(i=0;i<10;i++)
        {
            System.out.print(array[i]+" ");
        }
    }
}

```


Quicksort:

```
package es.studium.Quicksort;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Quicksort
{
    public static void main(String[] args) throws IOException
    {
        BufferedReader lectura = new BufferedReader(new InputStreamReader(System.in));
        int tabla[] = new int[10];
        int i;
        // Rellenar la tabla
        for(i=0;i<10;i++)
        {
            System.out.println("Dame el valor de la tabla ["+i+"]");
            tabla[i] = Integer.parseInt(lectura.readLine());
        }
        // Ordenar la tabla
        ordenar(tabla, 0, 9);
        // Mostrar los resultados
        System.out.println("La tabla ordenada sería:");
        for(i=0;i<10;i++)
        {
            System.out.print(tabla[i]+" ");
        }
    }
    public static void ordenar(int array[], int desde, int hasta)
    {
        int pivote;
        if(desde<hasta)
        {
            pivote = colocar(array, desde, hasta);
            ordenar(array, desde, pivote-1);
            ordenar(array, pivote+1, hasta);
        }
    }
    public static int colocar(int array[], int desde, int hasta)
    {
        int i, pivote, valor_pivote, temp;
        pivote = desde;
        valor_pivote=array[pivote];
        for(i=desde+1;i<=hasta;i++)
        {
            if(array[i] < valor_pivote)
            {
                pivote = pivote +1;
                temp = array[i];
                array[i] = array[pivote];
                array[pivote] = temp;
            }
        }
        temp = array[desde];
        array[desde] = array[pivote];
        array[pivote] = temp;
        return(pivote);
    }
}
```

Búsqueda de un elemento en una tabla ordenada

Búsqueda Binaria o Dicotómica:

```
package es.studium.Dicotomica;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Dicotomica
{
    public static void main(String[] args) throws IOException
    {
        BufferedReader lectura = new BufferedReader(new InputStreamReader(System.in));
        int tabla[] = new int[10];
        int i, elemento_buscado, inferior, superior, central, encontrado;
        // Rellenar la tabla
        for(i=0;i<10;i++)
        {
            System.out.println("Dame el valor de la tabla ["+i+"]");
            tabla[i] = Integer.parseInt(lectura.readLine());
        }
        System.out.println("Dame el valor a buscar");
        elemento_buscado = Integer.parseInt(lectura.readLine());
        // ORDENAR TABLA
        encontrado = -1;
        inferior = 0;
        superior = 8;
        while(inferior<=superior)
        {
            central = (inferior+superior)/2;
            if(tabla[central] == elemento_buscado)
            {
                encontrado = central;
                inferior = superior +1;
            }
            else
            {
                if(elemento_buscado<tabla[central])
                {
                    superior = central - 1;
                }
                else
                {
                    inferior = central + 1;
                }
            }
        }
        if(encontrado!=-1)
        {
            System.out.println("Se ha encontrado el elemento en la posición " +
encontrado);
        }
        else
        {
            System.out.println("NO se ha encontrado el elemento buscado");
        }
    }
}
```

Practicar

```
package es.studium.TallerTema3;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class TallerTema3
{
    static final int tam = 1000;
    public static void main(String[] args) throws IOException
    {
        // Variables
        int tabla[] = new int[tam];
        int tablaCopia1[] = new int[tam];
        int tablaCopia2[] = new int[tam];
        int tablaCopia3[] = new int[tam];
        int tablaCopia4[] = new int[tam];
        int numeroBuscado;
        int encontrado;
        long inicio, fin;
        // Apartado 1
        // Creación de una tabla de 1000 elementos de tipo entero entre -5000 y 5000
        tabla = generarTabla();
        mostrarTabla(tabla);
        // Apartado 2
        // Pedir número y buscar
        numeroBuscado = pedirNumero();
        inicio = Tiempo(); // Apartado 5
        encontrado = buscarNumero(tabla, numeroBuscado);
        fin = Tiempo(); // Apartado 5
        mostrarTiempo(inicio, fin); // Apartado 5
        if(encontrado!=-1)
        {
            System.out.println("Se ha encontrado el elemento en la posición " +
encontrado);
        }
        else
        {
            System.out.println("NO se ha encontrado el elemento buscado");
        }
        // Apartado 3
        tablaCopia1 = tabla;
        tablaCopia2 = tabla;
        tablaCopia3 = tabla;
        tablaCopia4 = tabla;
        // Intercambio
        inicio = Tiempo(); // Apartado 5
        tablaCopia1 = ordenarIntercambio(tablaCopia1);
        fin = Tiempo(); // Apartado 5
        mostrarTiempo(inicio, fin); // Apartado 5
        mostrarTabla(tablaCopia1);
        // Selección
        inicio = Tiempo(); // Apartado 5
        tablaCopia2 = ordenarSeleccion(tablaCopia2);
        fin = Tiempo(); // Apartado 5
        mostrarTiempo(inicio, fin); // Apartado 5
        mostrarTabla(tablaCopia2);
    }
}
```

```

// Burbuja
inicio = Tiempo(); // Apartado 5
tablaCopia3 = ordenarBurbuja(tablaCopia3);
fin = Tiempo(); // Apartado 5
mostrarTiempo(inicio, fin); // Apartado 5
mostrarTabla(tablaCopia3);
// Shell
inicio = Tiempo(); // Apartado 5
tablaCopia4 = ordenarShell(tablaCopia4);
fin = Tiempo(); // Apartado 5
mostrarTiempo(inicio, fin); // Apartado 5
mostrarTabla(tablaCopia4);
// Apartado 4
tabla = ordenarShell(tabla);
inicio = Tiempo(); // Apartado 5
encontrado = busquedaBinariaNumero(tabla, numeroBuscado);
fin = Tiempo(); // Apartado 5
mostrarTiempo(inicio, fin); // Apartado 5
if(encontrado!=-1)
{
    System.out.println("Se ha encontrado el elemento en la posición " +
encontrado);
}
else
{
    System.out.println("NO se ha encontrado el elemento buscado");
}
}
public static int[] generarTabla()
{
    int array[] = new int[tam];
    int i;
    for(i=0;i<tam;i++)
    {
        array[i] = (int) (Math.random()*10000-5000);
    }
    return array;
}
public static void mostrarTabla(int array[])
{
    int i;
    for(i=0;i<tam;i++)
    {
        System.out.print(array[i]+" ");
    }
    System.out.println("");
}
public static int pedirNumero() throws IOException
{
    BufferedReader lectura = new BufferedReader(new InputStreamReader(System.in));
    System.out.println("Dame el valor buscado: ");
    return Integer.parseInt(lectura.readLine());
}
public static int buscarNumero(int array[], int numero)
{
    int i, encontrado;
    i = 0;
    encontrado = -1;
    while((encontrado== -1) && (i<tam))
    {

```

```

        if(array[i]==numero)
        {
            encontrado = i;
        }
        i++;
    }
    return encontrado;
}

public static int[] ordenarIntercambio(int array[])
{
    int i, j, aux;
    for(i=0;i<tam-1;i++)
    {
        for(j=i+1;j<tam;j++)
        {
            if(array[i] > array[j])
            {
                aux = array[i];
                array[i] = array[j];
                array[j] = aux;
            }
        }
    }
    return array;
}

public static int[] ordenarSeleccion(int array[])
{
    int i, j, aux, indice_menor;
    for(i=0;i<tam-1;i++)
    {
        indice_menor = i;
        for(j=i+1;j<tam;j++)
        {
            if(array[j] < array[indice_menor])
            {
                indice_menor = j;
            }
        }
        aux = array[i];
        array[i] = array[indice_menor];
        array[indice_menor] = aux;
    }
    return array;
}

public static int[] ordenarBurbuja(int array[])
{
    int i, j, aux;
    for(i=0;i<tam-1;i++)
    {
        for(j=0;j<tam-1;j++)
        {
            if(array[j] > array[j+1])
            {
                aux = array[j];
                array[j] = array[j+1];
                array[j+1] = aux;
            }
        }
    }
    return array;
}

```

```

}
public static int[] ordenarShell(int array[])
{
    int i, aux, salto;
    boolean cambios;
    for(salto=5;salto!=0;salto=salto/2)
    {
        cambios = true;
        while(cambios==true)
        {
            cambios = false;
            for(i=salto;i<10;i++)
            {
                if(array[i-salto] > array[i])
                {
                    aux = array[i];
                    array[i] = array[i-salto];
                    array[i-salto] = aux;
                    cambios = true;
                }
            }
        }
    }
    return array;
}

public static int busquedaBinariaNumero(int array[], int elemento_buscado)
{
    int encontrado, inferior, superior, central;
    encontrado = -1;
    inferior = 0;
    superior = tam - 1;
    while(inferior<=superior)
    {
        central = (inferior+superior)/2;
        if(array[central] == elemento_buscado)
        {
            encontrado = central;
            inferior = superior + 1;
        }
        else
        {
            if(elemento_buscado<array[central])
            {
                superior = central - 1;
            }
            else
            {
                inferior = central + 1;
            }
        }
    }
    return encontrado;
}

public static long Tiempo()
{
    return System.currentTimeMillis();
}

public static void mostrarTiempo(long inicio, long fin)
{
    System.out.println("La tarea ha tardado "+ ( fin - inicio ) +" milisegundos");
}

```

}
}

09/08/2017



GRUPO

TUDIUM
FORMACIÓN

