



Código Asignatura:  
ISC-314

Nombre:  
Ronald Mariotti

Matricula:  
2014-0698

Trabajo:  
Quicksort

```

package logical;

import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class QuickSort {

    public static void main(String[] args) throws FileNotFoundException {
        int[] arr;

        long start1 = System.currentTimeMillis();
        arr = getArray();
        System.out.println("Usando el primer elemento de pivote :");
        System.out.println(quickSort(arr, 0, arr.length, new
PivotePrimerElemento()));
        long end1 = System.currentTimeMillis();
        System.out.println("Tiempo " + (start1 - end1) + " ms");

        long start2 = System.currentTimeMillis();
        arr = getArray();
        System.out.println("\nUsando el ultimo elemento de pivote :");
        System.out.println(quickSort(arr, 0, arr.length, new
PivoteUltimoElemento()));
        long end2 = System.currentTimeMillis();
        System.out.println("Tiempo " + (start2 - end2) + " ms");

        System.out.println("\nUsando la mediana de tres :");
        long start3 = System.currentTimeMillis();
        arr = getArray();
        System.out.println(quickSort(arr, 0, arr.length, new MedianaDeTres()));
        long end3 = System.currentTimeMillis();
        System.out.println("Tiempo " + (start3 - end3) + " ms");

    }

    private static int[] getArray() throws FileNotFoundException
    {
        Scanner in = new Scanner(new File("Inversion//QuickSort.txt"));
        try {
            int[] arr = new int[10000];
            for (int i = 0; i < 10000; i++) {
                arr[i] = in.nextInt();
            }
        }
    }

```

```

        return arr;
    } finally {
        in.close();
    }
}

private static int quickSort(int[] arr, int inicio, int fin, Pivote seleccion) {
    if (fin - inicio < 2)
    {
        return 0;
    }

    seleccion.mover(arr, inicio, fin);
    int pivotPosicion = particion(arr, inicio, fin);

    //ordenar de izq a derecha
    return fin - inicio - 1 + quickSort(arr, inicio, pivotPosicion, seleccion) +
    quickSort(arr, pivotPosicion + 1, fin, seleccion);
}

private interface Pivote {
    //Elige un pivote y lo mueve al inicio.
    void mover(int[] arr, int ini, int fin);
}

private static class PivotePrimerElemento implements Pivote {
    public void mover(int[] arr, int ini, int fin) {
        // El pivote es en el inicio aqui
    }
}

private static class PivoteUltimoElemento implements Pivote {
    public void mover(int[] arr, int ini, int fin) {
        int temp = arr[ini];
        arr[ini] = arr[fin - 1];
        arr[fin - 1] = temp;
    }
}

private static class MedianaDeTres implements Pivote {

    public void mover(int[] arr, int ini, int fin) {

        int mediana;

```

```

int medio = (ini + fin - 1) / 2;
if (arr[ini] < arr[medio])
{
    if (arr[medio] < arr[fin - 1])
    {
        mediana = medio;
    } else if (arr[ini] < arr[fin - 1])
    {
        mediana = fin - 1;
    } else
    {
        mediana = ini;
    }
} else
{
    if (arr[fin - 1] < arr[medio])
    {
        mediana = medio;
    } else if (arr[fin - 1] < arr[ini])
    {
        mediana = fin - 1;
    } else
    {
        mediana = ini;
    }
}

// Mueve la mediana al frente
int temp = arr[ini];
arr[ini] = arr[mediana];
arr[mediana] = temp;
}
}

```

```

private static int particion(int[] arr, int ini, int fin)
{
    int pivote = arr[ini];
    int pivotPosicion = ini;
    // Moviendo el intervalo para determinar la posicion del pivote
    for (int i = pivotPosicion + 1; i < fin; i++)
    {
        // moviendo lo que se menor que el pivote hacia el frente
        if (arr[i] < pivote)
        {

```

```
        int temp = arr[pivotPosicion + 1];  
        arr[pivotPosicion + 1] = arr[i];  
        arr[i] = temp;  
        pivotPosicion++;  
    }  
}  
arr[ini] = arr[pivotPosicion];  
arr[pivotPosicion] = pivote;  
return pivotPosicion;  
}  
  
}
```