|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación salas A y B

# Profesor: Asignatura:

*Grupo: No de Práctica(s): Integrante(s):*

Tista Garcia Edgar

Estructura de Datos y Algoritmos II

5

1

Calzada Martinez Jonathan Omar

# No. de Equipo de cómputo empleado

*Semestre:*

2019-2

7/02/2019

# Fecha de entrega: Obervaciones:

CALIFICACIÓN:

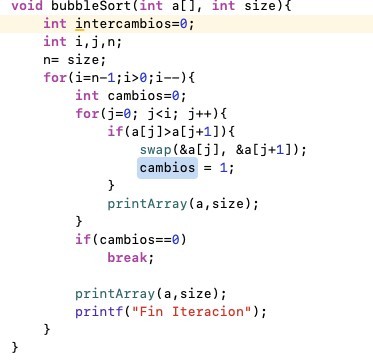
**Introduccion:**

Un algoritmo de ordenamiento es utilizado como su nombre lo dice a ordenar información, se puede hacer de mayor a menor o menor a mayor, existen diversas maneras para poder realizar los ordenamientos, pero dependiendo la cantidad de información a ordenar, así como hardware es como se elige al que mejor nos conviene ya sea por la cantidad de tiempo o la cantidad de recursos utilizados de la computadora.

Para esta practica estudiaremos el algoritmo de la Burbuja (BubbleSor ) y el algoritmo quickSor de orden rápido implementado distintas cantidades de información.

**Desarrollo:**

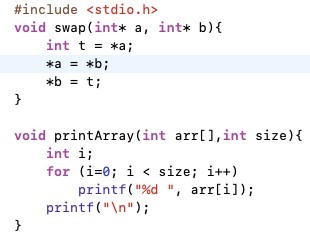
En la librerìa de ordenamiento se encuentran las funciones de ordenamientos quickSort y bublleSort, en ellas se encuentra el algoritmo correspondiente para ser llamado en la funcion principal segun se requiera o se necesite.



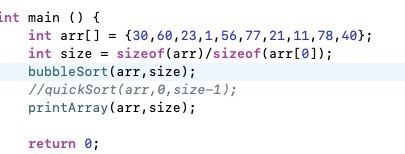
También en la biblioteca de ordenamiento encontramos una función llamada partición donde agarra el valor que se va a utilizar de pivote para empezar a hacer el ordenamiento.

Biblioteca de UTILIDADES

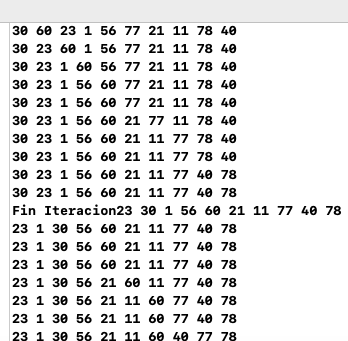
En la biblioteca de utilidades encontramos funciones que utilizan apuntadores para poder hacer los cambios de información en las variables a la hora de hacer los intercambios en los ordenamientos.

Así también imprime el los resultados obtenido de los ordenamientos ya sea utilizando el algoritmo de Burbuja o quickSor

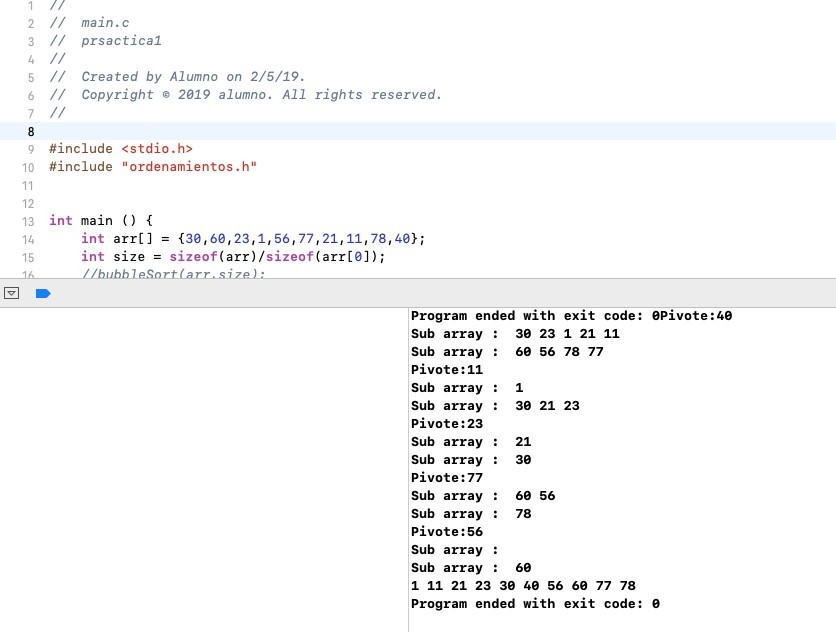
En la función principal encontramos un arreglo en donde lo inicializamos con valores arbitrarios dados por el profesor.



Utilizando el código con la función Burbuja obtenernos estos resultados



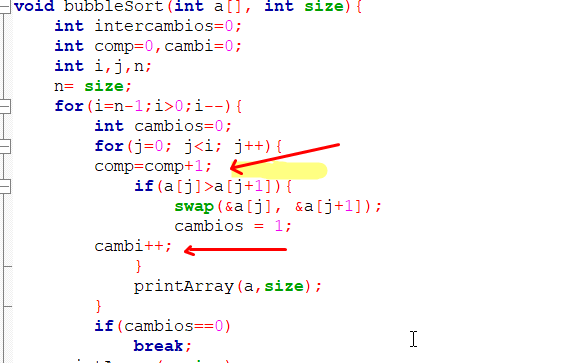
Utilizando la función quickSort obtenernos estos resultados



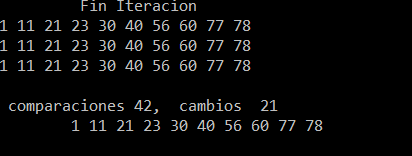
C) Agrega las instrucciones necesarias para contabilizar las comparaciones y los intercambios realizados

Para el algoritmo BubbleSort Se le agrego un variable llamada como para las contabilizaciones y una variable cambio que contabiliza los cambios hechos, colocados de la siguiente manera.

Esto se hizo con el arreglo proporcionado por el profesor.



Como resultado:



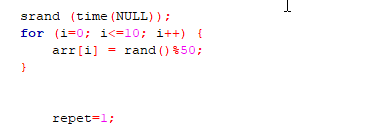
En total hubo 42 comparaciones y 21 cambios.

Para el algoritmo de quickSort

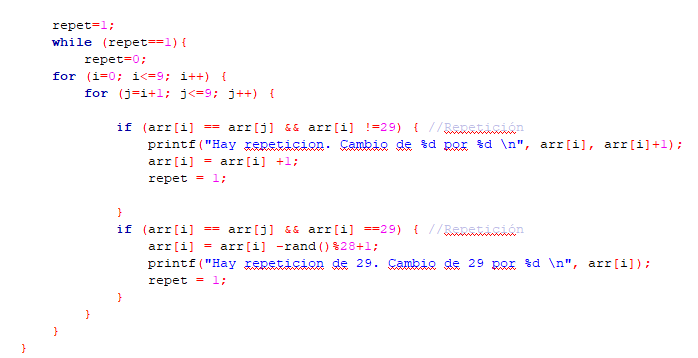
Insertar un contador en el algoritmo de quickSort no se pudo, en esta parte tengo que estudiar más ya que no se logro encontrar la manera de insertar el contador y así mismo que imprima la cantidad de comparaciones y cambios.

C.2) Realiza pruebas con arreglos de diferentes tamaños de entrada 10,50,100,500 elementos y verifica el crecimiento del numero de operaciones.

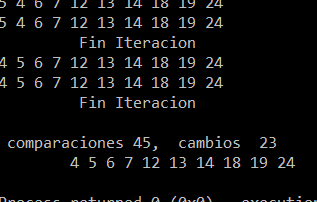
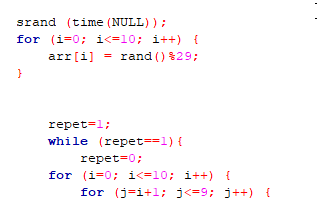
Para poder realizar el ejercicio con números aleatorios se utilizó la función srand para poner el rango de valores.



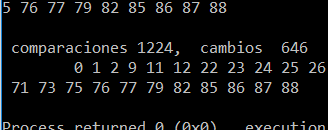
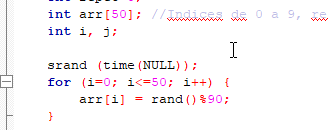
Después se implemento un while para detectar y cambiar un numero repetido. Se hizo de esta manera:



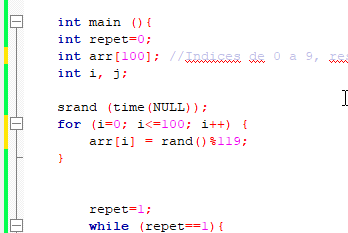
Realizando la prueba con 10 números para burbuja:

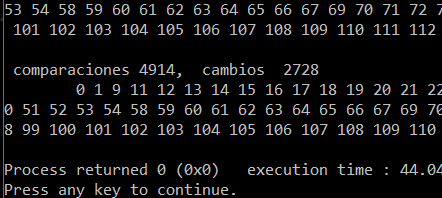


Realizando la prueba con 50 números para burbuja:



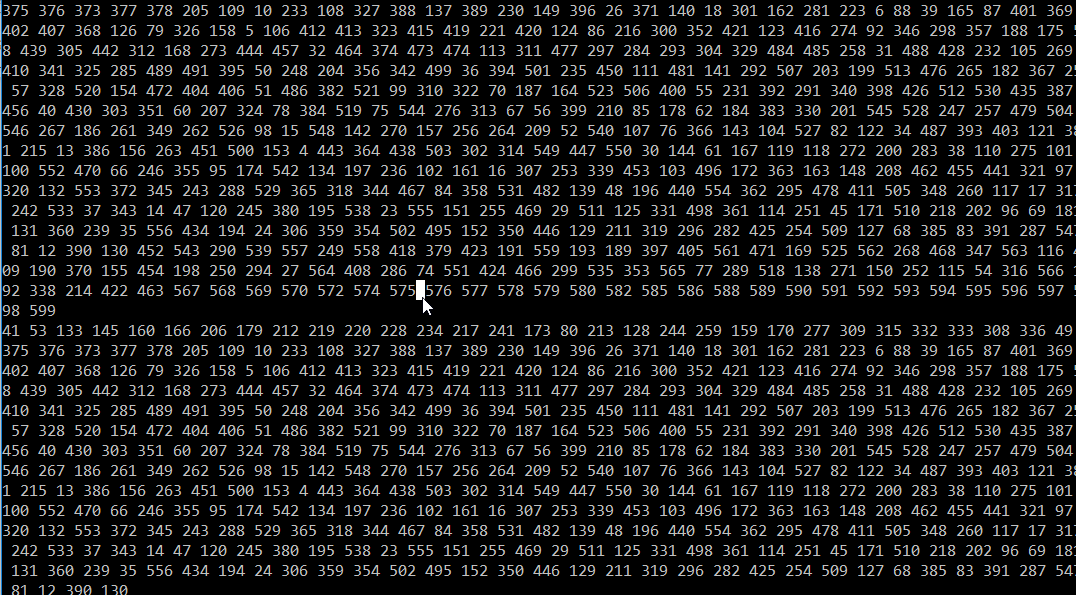
Realizando la prueba con 100 números para burbuja:



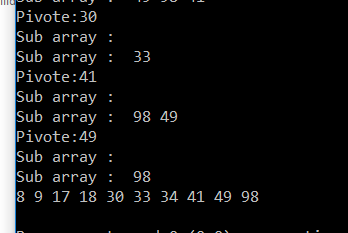


Realizando la prueba con 500 números para burbuja:

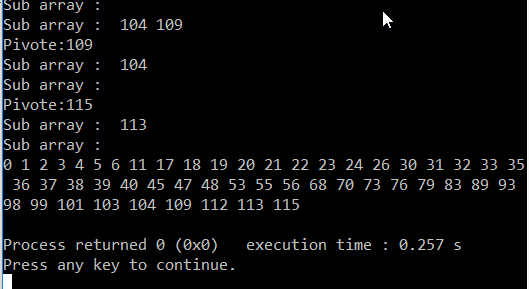
Para esta prueba debido a la complejidad el algoritmo Burbuja tardo más de 10 minutos haciendo el intercambio y cerre el ejecutable, la imagen es la siguiente.



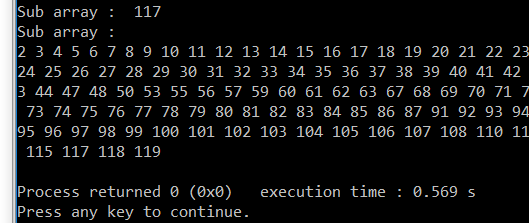
Realizando la prueba con 10 números para quickSort:



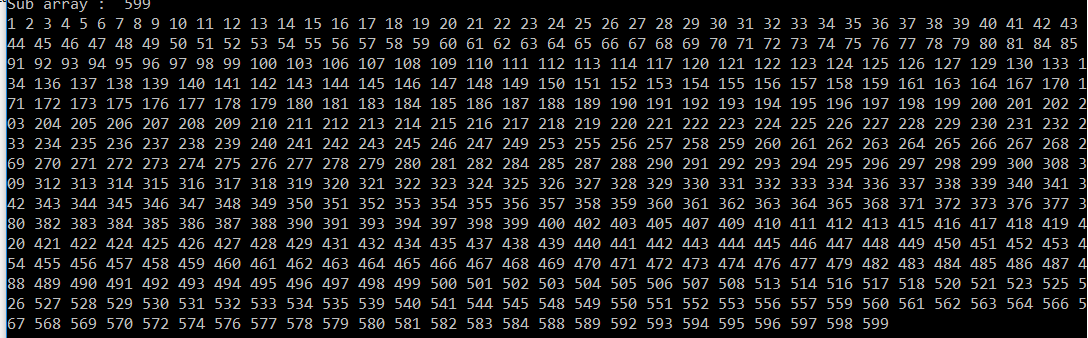
Realizando la prueba con 50 números para quickSor:



Realizando la prueba con 100 números para quickSort:

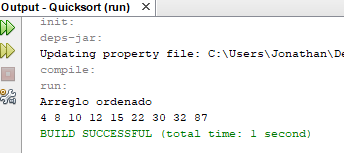
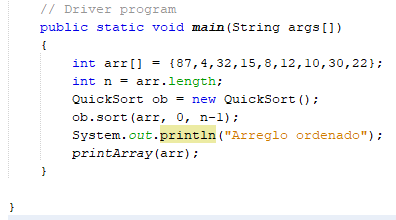


Realizando la prueba con 500 números para quickSort:



3) QuickSort -

Compila y ejecuta el proyecto.



Comentarios acerca del lenguaje Java:

Me doy cuneta que el lenguaje no cambia mucho. Muchas cosas son similares a C y a otros lenguajes como por ejemplo la declaración las variables el “for” y las asignaciones como i++; o arr[i]=temp; , así como también las funciones son parecidas a C.

Lo que si cambia aun que no mucho es la sintaxis al querer imprimir en pantalla, se le tiene que agregar System.out.printl, es un poco mas largo a comparación de C.

public static void main(String args[]) esta línea me causa un poquito de ruido porque aun no comprendo bien su uso.

Para compilar no hay diferencia solo los iconos que son diferentes ha y que la ejecución se hace en la parte baja del compilador de Netbeans.

Conclusiones:

Los algoritmos de ordenamiento tienen su grado de complejidad, así como ventajas y desventajas según sea el caso de los datos a ordenar como por ejemplo la cantidad de datos.

Para la mayoría de los casos siempre existe un algoritmo optimo que nos va a resolver el problema de la mejor manera.

El algoritmo de la burbuja su mejor caso es cuando la lista o los datos están ordenados, el peor de los casos es cuando no están ordenados son muchos datos. Por ejemplo, cuando hice la prueba para 500 números tardo más de 10 minutos en hacer el orden y mejor cerré la ejecución ya que era demasiado tiempo. Esto se debe a que trabaja con una complejidad de n^(2)

El algoritmo de quickSort en las pruebas fue muy rápido comparado con el de la Burbuja incluso en la prueba de 500 elementos.

La eficiencia de este algoritmo depende de la posición en donde se encuentre le pivote elegido. El mejor de los casos en cuando se encuentra en la mitad de los datos y el peor cuando se encuentra al último.

En java este algoritmo no cambio casi nada me refiero a las instrucciones y funciones ya que el algoritmo es el mismo.

Se cumplió el objetivo de la practica ya que se logró identificar la diferencia entre un algoritmo y otro, así como su complejidad, así como se logro entender la manera en como trabaja cada uno de los algoritmos vistos en la práctica.

Solo me hace falta estudiar la parte de el contador de comparaciones y cambios.