Métodos de ordenamiento

Estructura de datos – Universidad Tecnológica de Pereira

Estudiante José Luis Arciniegas Londoño – Ingeniería de sistemas

Los métodos de ordenamiento, son aquellos que nos permiten organizar datos en secuencia en la salida de un algoritmo analizado por dicho método, en estos encontramos a tres que son bastante importantes y que ayudan a organizar estos datos de manera correcta, los métodos de ordenamiento son los siguientes:

Bubblesort (Ordenamiento de burbuja).

Insertion Sort (Ordenamiento por inserción).

Quick Sort (Ordenamiento rápido).

Estos métodos encierran dentro de sí algoritmos que pueden ser de lo más sencillo, hasta lo más complejo, pero todos buscan un mismo objetivo que es el orden de los datos, pero entre todo esto ¿cuál es el mejor para ordenar y más eficiente?

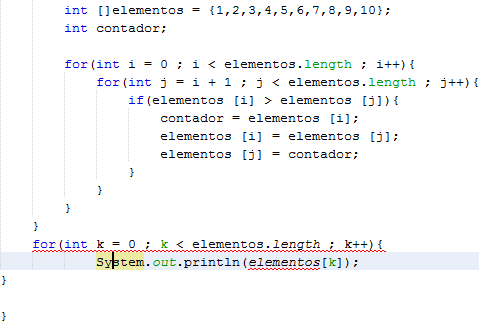
**Analicemos el Bubblesort:**

El Bubblesort es un algoritmo sencillo de ordenamiento que busca entre los elementos de una lista los datos a organizar, eso sí dependiendo de cómo queremos que sean organizados, dando un ejemplo de cómo podemos organizarlos, podemos hacerlo de la siguiente manera:

Digamos que tenemos el siguiente conjunto de elementos:

**A={1,2,3,4,5,6,7,8,9}**

El conjunto A de por si esta ordenado de menor a mayor, pero que tal si lo necesitamos al contrario, o si los datos no son los números del 1 al 9 si no números totalmente aleatorios, pues este algoritmo lo que hará es organizarnos como nosotros programemos los elementos, mirándolo desde Netbeans, en lenguaje de programación JAVA, conseguimos el siguiente algoritmo:



En este código, vemos que se necesitan de tres for, estos for son los que nos permiten hacer todo el escaneo de los elementos de dicho vector que declaramos anteriormente con el nombre de “elementos”, el primer for es el que nos facilitara el escaneo de todos los elementos del vector, el siguiente analizara el segundo elemento del vector para saber si es mayor al anterior y lo organizara, haciendo el proceso del for vemos que:

Escanea el número 1, que es el primer elemento de este vector y lo posiciona en la primera posición, y siguiente llama al segundo for que verifica la segunda posición del vector que es el número 2, si detecta que es mayor que el 1, entonces lo pone en la primera posición, este proceso se repite hasta que llega al número 10 y posiciona el vector de la siguiente manera:

**10,9,8,7,6,5,4,3,2,1**

Lo que permite escanear que cada número sea mayor al siguiente es el condicional que utilizamos, que debemos de pasar por cada uno de los for y verificar por medio del operador lógico mayor que “>”, si esto es correcto, entonces coloca el número que sea mayor en la primera posición, si detecta otro número mayor al siguiente mira si el primero es mayor o no, de no ser así lo posiciona en la segunda posición del vector.

El Bubblesort, es un algoritmo que puede llegar a programarse muy fácil si entendemos el concepto del ordenamiento de datos que queremos alcanzar, tenemos que tener en cuenta que el proceso puede ser bastante complejo dependiendo del número de datos que vallamos analizando con cada uno de los ciclos, por esto si tenemos miles de datos puede que demore un poco más de lo necesario, pero siempre cumplirá el objetivo de ordenamiento.

**Analicemos el Insertionsort:**

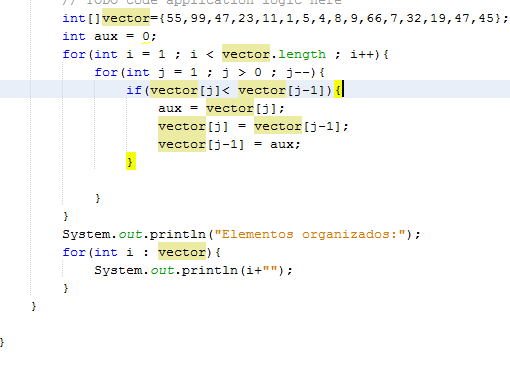
El Insertionsort es otro método de ordenamiento que en su código debemos de tener un poco más de experiencia en nuestro conocimiento del lenguaje, pero es mucho mas rápido y efectivo que el anterior Bubblesort, este tipo de ordenamiento se basa en como nosotros los seres humanos organizamos las cosas, ya sean de menor a mayor, o de mayor a menor, el caso es que la inserción empieza con un solo elemento, que este elemento obviamente está dentro de un conjunto de elementos, lo que hace este tipo de ordenamiento es mirar cuantos elementos hay en ese conjunto, y el empieza a comparar cada uno de estos elementos con el anterior, para revisar si es mejor, si este es menor al primero lo ubica en la primera casilla y sigue con el siguiente, si tenemos el siguiente conjunto:

**A={2,8,4,6}**

Este algoritmo pasara a organizarlo de la siguiente manera:

**A=[2,4,6,8}**

Miremos un algoritmo:



Se tiene un vector de diferentes tipos de elementos, muy distintos entre sí:

int[]vector={55,99,47,23,11,1,5,4,8,9,66,7,32,19,47,45}

Lo que hará este algoritmo es pasarse por cada uno de los elementos del vector, empezando desde el 55 hasta el 45, y mirara cual es el menor en ese recorrido, y el detecta al menor de todos que es el 1, este coge el número 1 y lo posiciona en la primera posición, pasando a ser este el menor por excelencia de ese vector, y así sucesivamente hasta organizarlos todos, como vemos en el código requerimos de solo 2 for anidados, uno cumple la función principal de escanear cada número de ese vector, y el otro se encargara de pasar el número menor a la primera posición del vector.

El Insertion sort puede llegar a ser una herramienta muy robusta para alcanzar este ordenamiento de manera un poco más sencilla, pero que requiere un poco mas de procesamiento, pero en sí, su estructura da para que el cálculo sea correcto, eso si dependiendo del número de datos que requiramos a organizar.

**Analicemos el QuickSort:**

Como su nombre nos lo indica es el método de ordenamiento rápido, este lo que hace es elegir la técnica de “Divide y vencerás”, este algoritmo funciona dela siguiente manera, busca el primer elemento de la fila y el ultimo, analiza si el primero es mayor o menor al último elemento, si el primero es mayor que el ultimo, lo que hace es pasarlo a la última fila, y al hacer este proceso lo que hará es pasar a la segunda posición y hacer el mismo procedimiento, si el número que está en la casilla número 2 es mayor que el que está en la penúltima casilla, los cambia de posición, y así sucesivamente hasta llegar al centro.

**¿Cuál de todos estos métodos es mejor?**

En mi opinión el mejor para hacer los métodos de ordenamiento y que sean entendibles para la persona que lo programa es el Insertionsort, debido a que este se adapta a un lenguaje más natural del ser humano a pesar que está dentro de un medio digital, el método de inserción, lo que hace es coger número por número y verificar si cada uno de ellos es mayor o menor al anterior para poder organizarlo, hasta llegar al punto que los números van a estar organizados de manera tal que no necesite hacer más cálculos, los tres tipos de ordenamiento, fueron hechos para el propósito de solucionar el ordenamiento de datos de manera más sencilla, si bien sabemos que la recursividad puede ser un método que se acerque más a una respuesta pronta, es una de las que más procesamiento necesita y puede llegar a tener un grado de complejidad muy elevado, si conocemos últimamente las nuevas generaciones de computadoras, veremos que el método de inserción puede ser procesado perfectamente, y puede hasta llegar a ser bastante rápido, el Insertionsort definitivamente podría llegar a ser uno de los elegidos por mí para el ordenamiento de datos.