

TEMA VIII ARCHIVOS (Ficheros)

8.1 Noción de archivo

8.2 Terminología: estructura jerárquica

8.3 Soportes secuenciales y direccionables

8.4 Organización de archivos

8.4.1 Organización SECUENCIAL

8.4.2 Organización DIRECTA

8.4.3 Organización SECUENCIAL INDEXADA

8.5 Operaciones sobre archivos

8.5.1 Creación o carga de un fichero

8.5.2 Reorganización de un fichero

8.5.3 Clasificación de un fichero

8.5.4 Destrucción de un fichero

8.5.5 Reunión o fusión de ficheros

8.5.6 Rotura o estallido de un fichero

8.5.7 Gestión de un fichero

Creación

Apertura

Cierre

Borrado

Ampliación

8.5.8 Mantenimiento

Consulta

Actualización (Alta, Baja, modificación).

Operaciones de Entrada/Salida.

8.6 Procesamiento de archivos secuenciales

8.6.1 Creación

8.6.2 Consulta

8.6.3 Actualización (Alta, Baja, modificación)

8.7 Procesamiento de archivos de acceso directo

8.7.1 Creación

8.7.2 Consulta

8.7.3 Actualización

8.8 Procesamiento de archivos con organización secuencial indexada

8.8.1 Creación

8.8.2 Consulta

8.8.3 Actualización

8.9 Ficheros de texto

8.1 Noción de archivo

Las estructuras de datos se almacenan en memoria principal, por eso también se conoce como almacenamiento primario o principal.

VENTAJAS

- El tiempo de acceso a los datos es muy pequeño porque accedemos a memoria principal.
- Se tarda lo mismo en acceder a cualquier dato de la estructura.

INCONVENIENTES

- La información de una estructura de datos (arrays) solo permanece en memoria durante el tiempo de ejecución del programa en el que está definida y siempre que el ordenador esté encendido.
- La memoria principal es cara y de tamaño limitado con lo que si necesito trabajar al mismo tiempo con una gran cantidad de datos, estos no pueden estar cargados a la vez en memoria.

Para solucionar estos inconvenientes aparecen los dispositivos de almacenamiento secundario. Estas unidades contienen datos organizados en una estructura denominada ficheros. Las VENTAJAS del almacenamiento secundario son:

- Los datos permanecen almacenados en el dispositivo secundario.
- Puedo tener almacenado en estos dispositivos gran cantidad de datos y luego fraccionarlos en unidades más

pequeñas para llevarlos a memoria principal y allí procesarlos.

INCONVENIENTES

– El acceso a los datos es mucho más lento porque para poder procesar los datos hay que transmitirlos del dispositivo a memoria y viceversa.

8.2 Terminología: estructura jerárquica

* Entidad: es algo que posee ciertos atributos o propiedades a las cuales se les puede asignar valores.

* Campo: unidad elemental de información que representa a un atributo de una entidad. Internamente un campo se almacena como un conjunto de bytes. Un campo se define por **[nombre, tamaño, tipo de datos]**. La longitud del campo suele ser fija. Un campo se puede subdividir en subcampos.

* Registro : Un registro es un conjunto de campos relacionados lógicamente que pueden ser tratados como una unidad en un programa. A un registro así definido se le denomina registro lógico. A nivel de programación el registro se define como un tipo record con un conjunto de campos.

Type <NOMBRE>:record

CAMP1: tipo

...

CAMPN:tipo

endrecord

Los registros pueden ser de longitud fija si todos los registros del mismo tipo tienen el mismo número de campos ó de longitud variable si pueden tener diferente número de campos. A estos últimos también se los denomina de partes variantes porque tendrían una parte fija y otra que varía en función del valor de un determinado campo.

* Archivo–fichero: Un fichero es un conjunto de registros relacionados entre sí y organizados de tal forma que puedo acceder a los datos de ese fichero para actualizar, borrar, insertar ... Cuando proceso un fichero, con lo que trabajo en realidad es con registros.

* Clave de fichero: Es un campo, conjunto de campos, que sirve para identificar un registro y distinguirlo del resto. Ha esta clave que es distinta para cada registro y que lo identifica se la denomina primaria y puede ser numérica, alfanumérica ...

* Registro físico: Cantidad de datos que se transfiere entre ó bloque: un dispositivo de almacenamiento secundario y memoria principal en una operación de E/S. Esta transferencia siempre es necesaria porque los datos están almacenados en los periféricos pero luego para su tratamiento tienen que ser llevados a memoria principal. Un bloque puede estar formado por uno o más registros lógicos y a su vez éstos pueden ocupar más de un bloque.

*Factor de bloqueo: Es el número de registros lógicos que puede contener como máximo un registro físico o bloque. El factor de bloqueo será >1 si el tamaño del registro lógico es menor que el del físico. Será <1 si el tamaño del registro lógico es mayor que el del físico y será $=1$ si los dos tienen el mismo tamaño.

Puesto que el bloque es la unidad de transferencia en operaciones de E/S por una parte, cuantos menos bloques tenga, menos operaciones de E/S tendrá que realizar para acceder a los datos de un fichero y por tanto tardaré menos tiempo. Desde el punto de vista del tiempo, cuanto mayor sea el factor de bloqueo menos tardaré en procesar los datos. Por otra parte, aunque lo ideal sería desde el punto de vista anterior, que hubiera un solo bloque físico y que todos los registros lógicos cupiesen dentro de ese bloque, en la práctica, sin embargo, el tamaño de un bloque no puede ser tan grande como queramos porque cuando lo queremos transmitir a memoria principal se transfiere a una parte de la misma. Si el tamaño de este buffer es muy grande ocupará demasiada memoria principal y no quedará memoria para procesar los datos, además, de que esta memoria es cara.

Al elegir el tamaño de un bloque hay que buscar el equilibrio entre los 2 aspectos anteriores para determinar cual es el factor de bloqueo lógico.

Ej. 10000 reg/fichero

Factor de bloqueo= 100 reg. lógico/bloque

tiempo de acceso = 1 s/bloque

1000 reg/fichero x 1 s/bloque = 100 seg

10 reg./bloque

* bases de datos: Conjunto de datos relacionados almacenados internamente en una colección de ficheros.

La organización de datos lógica (como lo ve el usuario) es una organización jerárquica. El nivel más alto es la base. La base de datos esta compuesta por ficheros, estos por registro, estos por campos y estos por caracteres.

8.3 Soportes secuenciales y direccionables

Soporte de un fichero es el medio físico o dispositivo donde se almacena ese fichero. Hay 2 tipos de soporte:

1.- secuenciales

2.- direccionables

1.- soporte secuencial

Es aquel en el que los registros se almacenan consecutivamente tal que para acceder a un registro determinado tengo que pasar por todos los registros colocados antes que él. La forma en que los coloco puede venir dada por el orden físico en el que fueron almacenados los registros o por el orden ascendente o descendente de alguna clave de esos registros. El soporte secuencial por excelencia es la cinta magnética.

2.- soporte direccionable

Para acceder a un registro no necesito pasar por los que están colocados delante de él sino que directamente calculo su dirección en el soporte y accedo a ella. En este tipo de soporte cada registro debe tener un campo clave a partir del cual calcula su dirección. El soporte direccionable más utilizado es el disco magnético. En un disco, una dirección vendrá dado por:

nº de volumen – nº de cilindro – nº de pista – nº de registro

8.4 Organización de archivos

La organización de un archivo viene definida por un método de organización y método de acceso. El **método de organización** es la técnica que se utiliza para coocar los registros en un dispositivo externo. El **método de acceso** es el que utilizan los programas para poder utilizar los datos del registro. Este método dependerá del de organización.

Según el soporte empleado para almacenar un fichero, hay 2 métodos de acceso:

- 1.- secuencial; para acceder a un registro tengo que pasar por todos los anteriores.
- 2.- directo; accedo directamente.

En el soporte de almacenamiento hay 3 tipos de organización:

- secuencial
- directa
- secuencial indexada

8.4.1 Organización secuencial

En esta organización los registros están almacenados consecutivamente según el orden en el que fueron grabados en el dispositivo externo cuando se creó el fichero.

Habrà una marca especial colocada al final del fichero, EOF, e indica el fin del mismo. Este tipo de organización es soportada en todos los tipos de dispositivos de almacenamiento secundario.

8.4.2 Organización directa

En un archivo con organización directa el orden físico que ocupa el registro no se corresponde con el orden lógico sino que por medio de una función hay que establecer la correspondencia entre la clave de cada registro y su posición en el dispositivo. A esta función se la conoce como la función de direccionamiento.

Un fichero con organización directa tiene que tener:

- Clave primaria ó campo que identifique a cada registro en el fichero.
- Una función de direccionamiento para establecer la correspondencia entre la clave de cada registro y su dirección que ocupa en el soporte.
- Tiene que estar almacenado en un dispositivo direccionable.

En la práctica lo que manejo para acceder a los datos de un registro no son direcciones absolutas (volumen, cilindro, pista, registro), sino que lo que manejamos son direcciones relativas del registro respecto del principio del archivo.

Para establecer la correspondencia entre direcciones relativas y direcciones absolutas podemos utilizar el método de las divisiones sucesivas:

Ej. Features of H.D.

2.- Se reserva al final del disco una serie de pista para sinonimos.

8.4.3

Indexada=Indice.

Esta organización se define por medio de dos características:

** Todos los registros del fichero guardan una secuencia a través de una clave, por tanto, todos los registros tendrán un campo identificador.

** Siempre que se deseen hacer accesos al azar será necesario investigar jerárquicamente unos índices.

En el momento de la creación, los registros se almacenan consecutivamente de forma compacta. A la vez se van creando unas áreas de excedentes que nos van a servir para las inserciones futuras. También se van generando unos registros especiales denominados índices.

Un fichero con organización secuencial indexada tiene 3 partes:

1.- Área primaria

Coloco los registros del fichero en el momento de la creación.

2.- Área de índices

Los índices son registros que también se denominan entradas que genera y mantiene el propio modo de acceso y que nos sirven para los accesos al azar. Hay diferentes niveles de índices.

* Índice de pista;

Es el de más bajo nivel y es obligatorio, siempre tiene que existir. Hay un índice de pista por cada cilindro ocupando la pista 0 de cada cilindro. En la 1ª entrada tendrá la clave y dirección del registro clave más alta de esa pista de ese cilindro, y la 2ª entrada contiene la clave y dirección del primer registro de la cadena de excedentes asociados a esa pista.

* Índice de cilindro;

Es el inmediatamente superior al índice de pistas y también es obligatorio. Suele estar contenido en el cilindro 0 y contendrá una entrada por cada cilindro con la clave más alta de los cilindros contenidos en ese cilindro.

* Índice maestro;

Es opcional. Lo utilizaremos cuando el índice de cilindros ocupe más de 4 pistas y tendrá una entrada por cada pista de índices de cilindros con la clave más alta de esa pista.

3.- Área de desbordamiento

Se utiliza para actualizaciones e inserciones futuras.

8.5.8

El mantenimiento de un fichero son las operaciones que sufre el fichero durante su vida. Son de 3 tipos:

- 1.- Consulta; sacar información del fichero
- 2.- Actualización; poner al día el fichero
- 3.- Operaciones de lectura/excritura

Consulta. tiene como finalidad dar información total o parcial de los datos almacenados en un fichero. esta información se presentará en un dispositivo de salida.

Actualización. tiene 3 partes:

- 1.- Alta: consiste en insertar o adicionar un nuevo registro al fichero. Los pasos a seguir son:
 - Comprobar que el registro que vamos insertar no existe.
 - Localizar la posicion en la que quiero insertarlo.
 - Grabarlo en dicha posicion.
- 2.- Baja: consiste en eliminar un registro de un fichero. lo primero que hay que comprobar es que ese registro existe en el fichero. Hay dos tipos de bajas:

**** Lógica;** no borro el registro físicamente sino que lo marca de manera especial lo que me indicará que está dado de baja y que por tanto, no lo puedo utilizar, sin embargo, el registro sigue grabado con lo que se espacio no podrá ser utilizado por otro registro. Cuando un fichero tiene un gran numero de bajas lógicas conviene reorganizarlo.

**** Física;** borramos físicamente el registro del soporte en el que esta almacenado.

- 3.- Operaciones E/S: se utilizan para transferencias de informacion entre la memoria y los perifericos.

leer (NOM_FICH,datos) donde queda almacenado lo que leo del fichero

escribir(NOM_FICH,datos) variable que contiene lo que escribo en el fichero

8.6

El inconveniente de archivo secuencial es que para acceder a un registro tienes que leer todos los anteriores, por eso se suele utilizar para fichero que sufran pocas modificaciones. En algunos lenguajes no se permite añadir nuevos registros o modificar datos en un fichero secuencial, sino que hay que crear uno nuevo.

8.6.1

La creación de un fichero secuencial es un proceso repetitivo, voy cargando consecutivamente cada registro hasta que encuentro la marca EOF;

1º Abrir el fichero

2º Leer los datos de cada registro

3º grabar el registro

4º Cerrar

Algoritmo creación_secuencial

inicio

abrir(FICH_NUEVO)

repetir

leer_pant Datos_del_registro

escribir(FICH_NUEVO,DATOS_REG)

escribir (Mas datos?)

leer_pant RESP

hasta_que resp=N

cerrar(FICH_NUEVO)

fin

8.6.2

La consulta o búsqueda en un fichero secuencial implica la lectura de todos los registros anteriores, por tanto, en un fichero de N registros habrá que hacer como máximo N lecturas y como mínimo 1. Así, la media de acceso es $(N+1)/2$ lo que supone que es muy lento.

algoritmo CONSULTA_TODO_FICHERO

inicio

abrir(FICH,IN_FILE)

mientras NOT EOF(FICH) hacer

leer (FICH,DATOS)

escribir_pant DATOS

fin_mientras

fin

algoritmo CONSULTA_ALEATORIA { con clave X}

inicio

abrir (FICH,IN_FILE)

mientras NOT EOF(FICH) y MARCA=TRUE

leer (FICH,REG)

Si reg.clave=x

entonces MARCA=FALSE

fin_si

fin_mientras

Si MARCA=FALSE

entonces escribir_opant REG

sino escribir('No encontrado')

fin_si

cerrar(FICH)

fin

8.6.3

Dar de **alta** un registro en un fichero secuencial es añadir a ese registro al final de dicho fichero. Como esto muchos lenguajes no lo permiten, lo que tendré que hacer es crear un nuevo fichero a partir del antiguo pero que contenga el nuevo fichero fichero dado de alta.

inicio

abrir (ENTRADA,IN_FILE)

abrir(SALIDA,OUT_FILE)

mientras NOT EOF(ENTRADA) hacer

leer (ENTRADA,REGSITRO)

escribir (SALIDA,REGISTRO)

fin_mientras

leer_pant REG_NUEVO

escribir(SALIDA,REG_NUEVO)

cerrar (ENTRADA)

cerrar (SALIDA)

fin

Para dar de **baja** a un registro en un fichero secuencial se le marca como borrado y más tarde, cuando reorganize el fichero, lo que tendré que hacer será cargar todos los registros en otro nuevo fichero excepto los que estan marcados como baja.

inicio

abrir (ENTRADA,IN_FILE)

abrir (SALIDA,OUT_FILE)

mientras NOR EOF(ENTRADA) hacer

leer (ENTRADA,REGISTRO)

escribir_pant(REGSITRO)

escribir(dar de baja al registro',regsitro)

leer resp

si RESP=s

reg.baja<-- *

fin_si

escribir (ENTRADA,REGSITRO)

fin_mientras

mientras NOT EOF(ENTRADA)

leer (ENTRADA,REGSITRO)

si reg.baja<>*

entonces escribir (SALIDA,REGISTRO)

fin_si

fin_mientras

Cerrar (ENTRADA)

Cerrar (SALIDA)

fin

Para modificar un registro tenemos que encontrar el registro a modificar, modificarlo y luego reescribirlo.

Algoritmo MODIFICAR_SECUENCIAL

inicio

abrir (FICH,IN_OUT_FILE)

MARCA <---- TRUE

mientras (NOT END OF FILE(FICH)) Y (MARCA=TRUE)

leer (FICH,REG)

Si REG_CLAVE=X

entonces escribir_pant REG

{modificar datos}

escribir (FICH,REG)

MARCA <---- FALSE

fin_si

fin_mientras

Si MARCA=TRUE

escribir ('REGISTRO DE CLAVE X NO ENCONTRADO')

fin_si

cerrar (FICH)

fin.

8.7 Procesamiento de archivos secuenciales

La ventaja de estos ficheros se da en procesos al azar ya que para acceder a un registro concreto dando su clave se accede directamente. El inconveniente es que aunque el acceso es rápido se necesita una función de direccionamiento. Otro inconveniente que puede haber es que si no reservamos espacio para todo el rango de claves posibles esto puede dar lugar a que se produzcan sinónimos.

8.7.1 Creación o carga

La carga es también un acceso al azar, se van introduciendo los diferentes registros y para cada uno se calcula con la función de direccionamiento cual es la posición en que tiene que almacenarse ese registro en el soporte.

Algoritmo CREAR_DIRECTO

inicio

abrir (FICH,OUT_FILE)

repetir

leer_pant REG

{calcular dirección de almacenamiento}

Si DIR_ALMACENAMIENTO=LIBRE

entonces escribir(FICH,REG)

sino {buscar espacio en el area de exceden

tes y grabarlo allí}

fin_si

escribir ('Deseas introducir más datos?')

leer RESP

hasta_que RESP='N' O RESP='n'

cerrar FICH

fin

8.7.2 Consulta

En un fichero con organización directa no tiene mucho sentido hacer una consulta de todos los archivos del fichero porque esta organización está pensada para hacerla al azar. Para hacer una consulta de uno o varios registros en concreto hay que dar los siguientes pasos:

- Dar la clave del registro buscado
- Calcular su dirección en el soporte.
- Leer el registro de esa dirección y comprobar si es el buscado, si no es así buscarlo en el area de excedentes.

8.7.3 Actualización (alta, baja y modificaciones)

ALTA, pasos:

- 1.– Dar la clave del registro que vamos a dar de alta.
- 2.– Mirar si esa clave pertenece al rango de claves permitido.
- 3.– Comprobar que no existe ya ese registro, si he dado la clave calculo la dirección.
- 4.– Si se cumplen las 2 condiciones anteriores escribo el registro en la dirección que le corresponda.

BAJA, pasos:

- 1.- Localizamos si el registro a dar de baja existe.
- 2.- Si existe y no está dado de baja pondremos el campo que indique que está dado de baja con la marca correspondiente.

MODIFICACION, pasos:

- 1.- Localizo el registro que voy a modificar.
- 2.- Modifico su contenido.
- 3.- Lo reescribo.

8.8 Procesamiento de archivos con organizacion

secuencial_indexada

En un archivo secuencial indexado hay un area de indices, un area de datos y otro area para los desbordamientos o excedentes, mientras el programa utiliza un fichero indexado los indices se guardan en memoria por lo que el proceso al azar será muy rápido mediante la investigación jerárquica de índices.

8.8.1 Creación

La carga siempre es secuencial, se deben proporcionar los registros en secuencia de clave, durante la creación los registros sólo se crean en el area de datos.

8.8.2 Consulta

La consulta puede ser de 3 tipos:

- 1º. Consulta de todo el fichero, iremos leyendo pista a pista todos los registros y por cada pista su cadena de excedentes.
- 2º. Consulta a partir de una clave, localizamos esa clave mediante la investigación de índices y a partir de ella realizamos un proceso secuencial igual al del caso anterior.
- 3º. Consulta al azar, de un registro en concreto, por investigación de índices.

8.8.3 Actualización (alta, baja y modificaciones)

ALTA

Las altas posteriores a la creación del fichero siempre se crearán en el área de excedentes localizando por los índices donde le corresponde.

BAJA

Localizo el registro, lo marco como borrado y actualizo si es necesario el área de índices.

MODIFICACION

Localizarlo, modificarlo y reescribirlo.

8.9 Ficheros de texto

Son un caso especial de ficheros secuenciales de ficheros secuenciales que lo que contienen son secuencias de caracteres, normalmente en estos ficheros también se incluye, además de la marca EOF, otras marcas especiales como EOLN ...

EJERCICIOS

1.- Dado un fichero CLIENTES, copiar a otro fichero todos aquellos registros con saldo negativo y visualizar en pantalla toda esta información: Formato:

DNI:string[8]

NOMBRE:string[40]

SALDO:real

1) Grabarlo en el fichero morosos.

2) Visualizar este fichero en pantalla.

2.- En una empresa se tiene un archivo en disco llamado factu-ras que debe cobrar la empresa a fin de mes. Cada registro del fichero corresponde a una factura y sus campos son:

NUM_CLIENTE: entero

IMPORTE_PAGAR:real

El fichero está clasificado por NUM_CLIENTE y pueden existir varios registros por cliente. Diseñar un programa que obtenga en pantalla una relación de todos los clientes con facturas pendientes, sacando para cada cliente el importe total que debe abonar y al final el importe total que ha de recibir la empresa.

3.- Dado un fichero en disco llamado alumnos cuyo formato del registro es:

CURSO: entero

NOMBRE: string[40]

NOTA_MEDIA: real

Obtener por pantalla una relación por cada curso, en la que aparezca: – número de curso

– número de alumnos de ese curso

– número de aprobados

– número de suspensos

El fichero está clasificado por curso.

4.- Tenemos dos ficheros, EMPLEADOS y BAJAS con sus respectivos formatos:

"empleados" "bajas"

NUM_EMPLEADO: entero NUM_EMPLEADO:entero

NOM_EMPLEADO:string[40]

Obtener el fichero actual. El fichero bajas está depurado, es decir, todos los registros tienen su correspondiente en el fichero empleado. Los ficheros baja y empleados están clasificados por NUM_EMPLEADO.

5.- Sea un fichero llamado facturas que contiene información de las facturas mensuales de una empresa y sabiendo que contiene la siguiente información:

tipo FACTURA: record

NUM_CLIENTE: entero

FECHA:fecha

NUM_VENDEDOR:entero

IMP_BRUTO:real

segun_sea FORMA_PAGO

contado;DESCUENTO:entero

giro;BANCO:entero

SUCURSAL:entero

CUENTA:array[1..40] of char

fin_segun_sea

end_record

Este fichero esta clasificado por NUM_CLIENTE. Obtener una informacion por pantalla en la que aparezca por cada cliente el importe a pagar. Sacar otra relación con la cifra de ventas de cada vendedor los cuales están numerados de 1 a 9

6.- Un fichero TEMPERATURAS donde se recogen las temperaturas de una determinada ciudad a lo largo del año, existe un registro por mes con el siguiente formato:

NUM_DIAS: entero(28..31)

T_DIARIA: array [1..31,1..2] de entero

Se pide:

1) obtener la Temperatura media del año

2) obtener la Temperatura máxima y mínima del año con el mes y día que se produjo

3) repetir iterativamente un proceso que solicite por pantalla un día y un mes del año y visualice la Temperatura máxima, mínima y media de ese día. El proceso termina cuando el mes introducido sea 0.

7.- Una empresa mantiene un fichero histórico anual con los sueldos pagados a sus trabajadores, el registro tiene la siguiente estructura:

COD_DPTO: entero

COD_EMP: entero

NOM_EMP: string[40]

CATEG_EMP: entero

SUELDO: real

El fichero empleado está clasificado por COD_DPTO y dentro de cada departamento por COD_EMP. Diseñar un algoritmo que de la siguiente información:

a) Resumen de cada DPTO, del total de empleados y del sueldo media anual de dicho departamento.

b) Resumen por categoría profesional del total de empleados y del sueldo medio anual de dicho departamento.

c) Resumen por categoría profesional del total de empleados y del sueldo medio anual de esa categoría.

En la empresa hay 12 categorías numeradas del 0 al 11.

8.- Dado un fichero almacenes con el siguiente formato:

COD_ALMACEN: entero

COD_PRODUCTO: entero

STOK: entero

PRECIO_UNIDAD: real

El fichero está clasificado por el COD_ALMACEN, se trata de obtener el coste total de todos los productos en cada almacén.

TEMA 9 ORDENACION, BUSQUEDA E INTERCALACION

9.1 Introduccion

9.2 Ordenacion

9.2.1 Método de la burbuja ó intercambio

9.2.2 Ordenación por inserción

9.2.3 Ordenación por selección

9.2.4 Método de Shell

9.2.5 Quicksort

9.3 Búsqueda

9.3.1 Búsqueda secuencial

9.3.2 " binaria o dicotómica

9.3.3 " por conversión de claves, HASHING

9.4 Intercalación o mezcla

\-----\

9.1 Introducción

Para conseguir mayor eficiencia en el tratamiento de la información, tanto a nivel de almacenamiento interno (arrays) como externo (ficheros), muchas veces conviene tener la información ordenada. Por ejemplo puede ser útil en procesos de búsqueda.

9.2 Ordenación

Ordenación o clasificación es la operación que consiste en organizar un conjunto de datos en un orden determinado:

- Ascendente/Descendente; refiriendonos a números
- Alfabético; refiriendonos a caracteres

Hay 2 tipos de ordenación:

- 1.– Interna; clasificamos un array. Se denomina así porque se hace directamente en memoria con lo que se hará más rápido.
- 2.– Externa; clasificamos un registro de un archivo almacenado en un dispositivo externo más lento.

La **complejidad** de un algoritmo es la cantidad de trabajo realizado y se mide por el número de operaciones básicas que se han llevado a cabo. Para calcularla se analizarán el mejor, peor y medio de los casos posibles.

9.2.1 Método de la burbuja o intercambio

Su filosofía es ir comparando los elementos del array 2 a 2, intercambiando sus posiciones si es necesario y repetir este proceso hasta que ya tengo todo el array ordenado. Pasos del Algoritmo:

Tenemos almacenado en memoria un array A que números, el algoritmo funciona de la siguiente forma:

1º Comparamos A1 y A2 y los colocamos en el orden que deseemos. Comparamos A2 y A3 y hacemos lo mismo, realizamos esta operación hasta conseguir A n-1 con An.

Este primer paso implica n-1 comparaciones y despues de este paso el elemento mayor del array se posiciona hacia la última posición.

2º Repetir el paso primero, pero con una ordenación menos, es decir, terminará al compara A n-2 con A n-1 y obtendremos el segundo elemento mayor en A n-1.

3º Repetir el paso primero hasta comparar A n-3 con A n-2 y así sucesivamente hasta realizar el paso A n-1 en el que compararíamos A 1 y A 2 solamente.

Ej.

20 3 80 3 10 3 30

PASO1 PASO2 PASO3

1) 20 80 10 30 1) 10 20 30 80 1) 10 20 30 80

2) 20 10 80 30 2) 10 20 30 80

3) 20 10 30 80

Vemos que el proceso es una estructura repetitiva de pasos y comparaciones: bucles.

METODO 1

El algoritmo menos eficiente sería aquel que por cada paso realizase N-1 comparaciones, es decir, que no tuviese en cuenta que en cada paso ya queda colocado en su posición un elemento.

Algoritmo BURBUJA_1

inicio

desde I=1 hasta (N-1) hacer

desde J=1 hasta (N-1) hacer

Si $X(j) > X(j+1)$

entonces

AUX <----- $X(j)$

$X(j)$ <---- $X(j+1)$

$X(j+1)$ <-- AUX

fin_si

fin_desde

fin_desde

fin

METODO 2

Podemos introducir una mejora del método anterior teniendo en cuenta al hecho de que ya queda colocado un elemento en su posición correcta. Estando en un paso I bastará que realice $N-I$ comparaciones.

Algoritmo BURBUJA_2

inicio

desde I=1 hasta(N-I) hacer

desde J=1 hasta (N-I) hacer

Si $X(j) > X(j+1)$

entonces

AUX $\leftarrow$ X(j)

X(j) $\leftarrow$ X(j+1)

X(j+1) $\leftarrow$ AUX

fin_si

fin_desde

fin_desde

fin

La complejidad viene dada por la formula: