TEMA 19

Estructura y almacenamiento de datos. Tipos de ficheros. Métodos de acceso. Las bases de datos y hojas de cálculo. El tratamiento de textos.

Introducción

Veremos nociones básicas de los principales tipos de datos y estructuras. Algoritmos de ordenación y búsqueda.

1. Tipos abstractos y estructuras de datos

Los Tipos Abstractos de Datos (TAD) son una formalización de los elementos necesarios para generar una estructura adecuada. No dependen del lenguaje de programación. Son tipos de datos de alto nivel que solo indican un concepto.

Son modelos matemáticos utilizados para resolver problemas por parte de los programadores a alto nivel.

Independiza las formas de resolver los problemas de los lenguajes de programación y del hardware, centrándose en los algoritmos de resolución a alto nivel.

Ejemplos de TAD: matriz matemática y las colas.

Operaciones:

- **Creación**: definir y generar los TAD y sus elementos.
- **Transformación**: modificar los valores de los TAD.
- **Análisis**: obtener información de los TAD.

Estructuras utilizadas: tablas, árboles, listas, conjuntos, mapas y diccionarios.

1.1. Conjunto

Colección de valores **no ordenados**, no se permiten elementos repetidos.

Su funcionalidad es comprobar si algún elemento pertenece o no al conjunto. Sirven para clasificar valores por categorías.

1.2. Mapa

Es un conjunto de claves y un grupo de valores. **No ordenados**. Las claves no pueden estar repetidas y los valores si.

1.3. Diccionario

Es un mapa donde se le da un orden interno a las claves. **Ordenado**.

1.4. Tabla

Estructura de datos homogénea, tiene que cumplir las siguientes premisas:

- Componentes del mismo tipo
- Número predefinido de componentes, no se pueden modificar en tiempo de ejecución.
- Los elementos contienen una clave.

El uso más habitual de la tabla es localizar elementos.

1.5. Listas

Deben cumplir:

- Todos los componentes son del mismo tipo.
- Cada elemento va seguido de otro o de ninguno.
- Se utiliza algún tipo de orden para almacenar sus componentes.

Los elementos se insertan de forma **ordenada**.

Listas densas determinan su estructura por los propios elementos que la conforman, las listas enlazadas cada elemento contiene la información necesaria para llegar al siguiente.

Tipo de ordenamiento LIFO (pilas) Last In First Out y FIFO (Colas) First in First Out.

Operaciones: Búsquedas, ordenaciones, etc.

1.6. Árboles

Disponemos de nodos que representarían el tipo de datos y de niveles. Estructura dinámica, puede ser modificada en tiempo de ejecución, puede recorrerse en profundidad y amplitud.

Terminología:

- **Antecesor directo o padre**: nodo inmediatamente superior conectado.
- **Antecesor**: cualquier nodo superior.
- **Sucesor directo o hijo**: nodo inmediatamente inferior conectado.
- **Sucesor**: cualquier nodo inferior.
- **Nodo rama**: nodo con algún hijo.
- **Nodo hoja**: nodo sin hijos.
- **Nivel de un nodo**: Número de antecesores que tiene.
- **Grado de un nodo**: número de hijos.
- **Grado de un árbol**: el mayor de los grados de nodo.
- **Altura de un árbol**: el mayor nivel.

Tipos de árboles:

- **Binario**: cada nodo tiene un máximo de dos hijos.
- **Multirrama**: no tiene límite en el número de hijos de un nodo.
- **Balanceado**: entre todos sus nodos hoja no hay una diferencia mayor a un nivel.

Operaciones: recorrer el árbol, insertar elementos, etc.

3 métodos para recorrer los árboles:

- **Pre-orden**:
  - Visitar la raíz.

    - Recorrer subárbol izquierdo
    - Recorrer subárbol derecho.
- **In-orden**:
    - Recorrer subárbol izquierdo.
    - Visitar la raíz.
    - Recorrer subárbol derecho.
- **Post-orden**:
    - Recorrer subárbol derecho.
    - Recorrer subárbol izquierdo.
    - Visitar la raíz.

2. Almacenamiento de datos

Sistemas de memoria secundaria para dotar a esta de carácter indefinido y que no se vea afectada por los diferentes procesos que concurran en el procesador y la memoria principal.

Acceso al máximo de la capacidad de almacenamiento y optimizar la velocidad de acceso a la información.

El tamaño de los ficheros puede cambiar.

Se guardaban los datos en cintas magnéticas con **acceso secuencial**.

Los discos duros, **acceso por índices**.

Estructura de árbol con carpetas y ficheros, dando lugar a la estructura de directorios.

El orden de los registros, sigue **varias estructuras tipo**:

- Ficheros ordenados.
- Ficheros desordenados.
- Ficheros dispersos o hashing.
- Agrupamiento o clustering.

Los **bloques** son elementos lógicos de almacenamiento y se puede definir su tamaño por el usuario.

Clúster es la unidad de espacio más pequeña que se puede asignar a un fichero.

Tiempos y características:

- **Tiempo de búsqueda**: tiempo que se tarda en encontrar el primer dato que hay que leer.
- **Velocidad de rotación**: velocidad de transferencia de la información y dependerá de la velocidad de giro del disco.
- **Tiempo de latencia**: tiempo de espera desde que se posiciona en la pista concreta hasta que llega al sector deseado.

3. Métodos de acceso a los datos

Operaciones:

- Petición de acceso a los datos por algún programa al S.O.
- El S.O. realiza la petición al Gestor de Ficheros que comprueba que la petición es correcta y comienza a acceder al fichero.
- Buscar el sector/bloque correspondiente en la FAT.
- Se realiza la lectura del sector y se almacena en la RAM para hacerla accesible al procesador.
- El procesador de E/S se encarga de acceder correctamente al disco y por último da la orden al controlador correspondiente para llevar a cabo la escritura/lectura.

3.1. Tipos de ficheros

- **Ficheros ordenados**: se van colocando según el campo de ordenación. Se suelen utilizar para hacer índices. Operaciones:
  - **Buscar**: búsqueda binaria o búsqueda lineal.
  - **Leer**: Lectura ordenada, por el campo de ordenación.
  - **Insertar**: buscar la posición, abrir el hueco necesario y guardar los datos.
  - **Eliminar**: encontrar el registro y eliminarlo.
  - **Modificar**: dependerá de si la modificación quepa donde está ubicado o no.
- **Ficheros desordenados**: se insertan según se van generando las peticiones. Para almacenar datos que se procesarán más tarde. Operaciones:
  - **Buscar**: búsqueda lineal.
  - **Leer**: se necesita una ordenación externa.
  - **Insertar**: se añade por el final.
  - **Eliminar**: encontrar el registro y eliminarlo.
  - **Modificar**: igual que antes.
- **Ficheros dispersos**: la dirección de los registros se obtiene aplicando una función sobre uno de los campos, llamado campo de dispersión. Tipos de dispersión: estática, dinámica, extensible y lineal. Resuelve la función para gestionar las colisiones y guardado correcto de los datos.

3.2. Algoritmos (para optimizar métodos de acceso a los datos)

Es una secuencia de instrucciones ordenadas que representa un modelo de solución a un problema determinado.

Representar los algoritmos:

- **Lenguaje de alto nivel**: secuencia de instrucciones sin tener en cuenta la forma de llevarlas a cabo.
- **Lenguaje de programación**: es como se implementa el algoritmo.
- **Diagrama de flujo**: representación gráfica que ayuda a su depuración y optimización.

Características:

- **Preciso**: los pasos e instrucciones sin ambigüedades ni interpretaciones.
- **Finito**: debe terminar en algún momento.
- Debe obtenerse un **resultado**.

Pasos para **generar un algoritmo de cómputo**:

- **Análisis del problema a resolver**: buscar la solución más eficaz.
- **Diseño de diagrama de flujo**: visualizar el algoritmo de forma gráfica ayuda a su implementación y optimizado.
- **Definición en lenguaje de alto nivel**: determinan las instrucciones en lenguaje de alto nivel (español). Independizar la forma de solucionar el problema con el lenguaje de programación.
- **Prueba de algoritmo**: se comprueba que el algoritmo resuelve el problema y que siempre se obtiene la misma solución para los mismos datos.
- **Implementación**: Se traslada el algoritmo al lenguaje de programación.

Complejidad algorítmica es la cantidad de recursos que necesita el algoritmo para resolver un problema. Se puede medir en espacio o en tiempo.

3.3. Algoritmos de búsqueda y ordenación

Algoritmos recursivos y algoritmos iterativos.

Los algoritmos de ordenación pueden ordenar los ficheros de forma interna o externa.

Algoritmos de ordenación:

- **Selección**: se busca el elemento más pequeño y se coloca en primera posición, se repite para buscar el siguiente más pequeño y colocarlo a continuación y así sucesivamente.
- **Burbuja**: se buscan pares de elementos adyacentes y se comparan entre sí ordenándolos hasta que se tiene todo completo.
- **Inserción directa**: se dispone de una parte de los datos (sublista) ordenados y se van insertando los nuevos elementos en el sitio correspondiente para no perder el orden establecido. Esa parte de datos se hace cada vez mayor hasta que se completa el ordenamiento.
- **Inserción Binaria**: igual que el anterior, utiliza un sistema de búsqueda binaria.
- **SHELL**: se utiliza para listas con muchos elementos. Se compara con elementos distantes cierto espacio para ir acotando el lugar en el que insertar. Posteriormente se van realizando pasadas disminuyendo la distancia, hasta que esta se hace 1 y se comprueba que toda la lista está ordenada.
- **Ordenación rápida (Quicksort)**: se utiliza un elemento llamado pivote, de forma que se generan dos sublistas. Una a la izquierda

con los elementos menores que el pivote y otra a la derecha con los elementos mayores. Se aplica el algoritmo a cada una de estas sublistas para llevar a cabo el ordenamiento completo.
- **Binsort**: clasificación por urnas. Es necesario conocer algo sobre las claves a ordenar. Los elementos se sitúan en urnas según la clave que tengan. Se tienen tantas urnas como claves y los elementos se van introduciendo en ellas. Si hay varios elementos con la misma clave se ordenan con algún tipo de índice.
- **RadixSort**: uso de urnas. Ordenar por dígitos o caracteres. Se coge el dígito de menos peso y se organizan los elementos en torno a él introduciendo en urnas todos los que tengan el mismo dígito. Luego se lleva a cabo el proceso con el siguiente dígito y así sucesivamente. Cuando lleguemos al último dígito, quedará enlazar todas las urnas para que la lista quede ordenada.

Algoritmos de búsqueda:

- **Secuencial**: recorrer todos los elementos hasta encontrar el o los que buscamos.
- **Búsqueda binaria**: se debe de partir de una tabla o lista ordenada. Se divide en dos trozos y se comprueba el elemento. Si es mayor debe estar en la parte que contiene los elementos menores y si es menor, en la lista que contiene los elementos mayores. Posteriormente ese trozo se vuelve a dividir y así sucesivamente hasta que lleguemos al elemento deseado.
- **Búsquedas basadas en tablas Hash**: se asigna a cada elemento un índice obtenido mediante una función Hash. Esa función aporta un dato que permita identificar y ordenar el elemento. Luego se busca en la tabla de índices. Hay que tener en cuenta las posibles colisiones.

Resolución de colisiones:

- **Encadenamiento separado o Hashing abierto**: se construye para cada clave que salga una tabla. Se suele usar una LIFO para ir guardando los elementos.
- **Direccionamiento abierto o Hashing cerrado**: se usa un vector en el que se pone una clave en cada una de las casillas. Se utiliza el rehashing, que es una función para determinar el elemento exacto, una vez se ha localizado la clave que se busca.

3.4. Resumen de métodos de acceso y tipos de ficheros

Procesamiento de los ficheros (la forma de acceder a los datos en el disco duro):

- **Secuencial**: los ficheros se tienen que leer uno detrás de otro y hay que pasar por todos hasta llegar al deseado.
- **Directo**: se puede acceder directamente a los datos sin necesidad de leer todos los anteriores.

Organización de ficheros:

- **Secuencial**: uno detrás de otro.
- **Directo**: se determina posición por algún algoritmo.
- **Indexado**: se disponen secuencialmente, pero se llevan a cabo índices mediante algoritmos.

Tipología de ficheros:

- **Ficheros lineales simples**: los registros se disponen uno detrás de otro.
  - Apropiados para procesamientos secuenciales.
  - Para modificarlos es necesario hacer copias de los ficheros.
  - Lento para consultas puntuales.
  - Aprovecha al máximo la capacidad.
- **Ficheros lineales encadenados**: los datos son lineales, van uno detrás de otro, pero usan una clave para poder facilitar el acceso a los mismos.
  - Se usan listas simples y múltiples, anillos, árboles, etc.
  - Los registros deben contener un campo extra con el puntero.
  - Problemas de aparición de huecos cuando se eliminan ficheros. Procesos de recuperación de huecos y gestión dinámica de espacio libre.
- **Ficheros con índices (indexados)**: similares a los lineales encadenados, al fichero le acompaña un fichero de índice que permite el acceso directo a los datos.
  - El índice debe contener clave y puntero.
  - Índice total (denso) cuando apunta a un registro de ficheros e índice escaso (no denso) cuando ese registro al que apunta el índice deba estar ordenado para facilitar el acceso a los datos.
  - Mejora mucho el acceso a los datos.

4. Formatos de información y ficheros

La mayoría de las aplicaciones guardan la información en ficheros a los cuales les suelen dar un formato propio. Para garantizar que esa información será completamente accesible para el usuario y poder ser editada por este manteniendo todas sus características.

4.1. Ficheros con información únicamente de texto

- .**TXT**: textos planos que no tienen formato.
- .**RTF**: textos con negrita, cursiva o subrayado.
- .**DOC** o .**DOCX**: de Word de Microsoft.
- .**ODT**: de OpenOffice y LibreOffice.

4.2. Ficheros con información de imagen

- .**JPG**:
- .**TIFF**: aumenta la calidad de los JPG.
- .**GIF**: animaciones.
- .**PNG**

4.3. Ficheros con información diversa

Puede haber elementos mixtos con texto, imágenes, formatos de texto y otros elementos.

- .**PS**: PostScript, se ve el fichero tal y como se imprimirá.
- .**PDF**
- .**swf**: Shockwave Flash contiene animaciones y gráficos vectoriales multimedia.

4.4. Ficheros con información de datos

Para trabajar con datos como tablas.

- .**csv**: ficheros separados por comas.
- .**xml** o **xls**: hoja de cálculo.
- .**ods**: hoja de cálculo de OpenOffice.

4.5. Ficheros con información comprimida

Para ahorrar espacio de almacenamiento. Se necesita un proceso de comprensión y antes de volver a utilizar el fichero hay que descomprimirlo.

- .**ZIP**
- .**RAR**

4.6. Tipos de fichero en función de su uso

Los ficheros pueden ser permanentes o temporales.

Ficheros permanentes:

- **Maestros**: estado actual de los ficheros que puedan ser modificados.
- **Constantes**: datos que no cambian
- **Históricos**: información pasada que pueda resultar relevante.

Los ficheros temporales por poco tiempo y suelen servir de puente para realizar modificaciones en los archivos permanentes.

4.7. Denominación de ficheros.

Precaución con las tildes, ñ y caracteres especiales.

5. Bases de datos y hojas de cálculo

5.1. Bases de datos

Es un conjunto de elementos que se almacenan de forma ordenada para su posterior consulta por parte del usuario.

Dos tipos: **relacionales** y de **datos orientadas a objetos**.

5.1.1. Sistemas de gestión de bases de datos relacionales. Características y componentes

Sistema de base de datos es el que nos proporciona la capacidad de tener un control centralizado de los datos e información disponibles.

Características:

- **Reducción de información innecesaria**: al estar centralizada sin necesidad de duplicarla. Requiere de controles y permisos.
- **Evitar inconsistencia**: ser capaz de encontrar los datos contradictorios.
- **Seguridad**: permisos de acceso.
- **Independencia de los datos**: se hace independiente la información de los programas utilizados para mostrarla.

Se deben crear complejas estructuras de datos manteniendo la consistencia necesaria para manejar grandes cantidades de datos.

Niveles de abstracción:

- **Nivel físico**: modelar cómo se almacenan físicamente los datos. Usando esquemas y definir el significado de cada bit.
- **Nivel conceptual**: modela los datos que se almacenan en la Base de Datos y las relaciones entre ellos.
- **Nivel lógico**: nivel de abstracción más cercano al usuario, modelar la forma en que estos acceden, visualizan y modifican los datos existentes en la Base de Datos.

Tipos de bases de datos:

- **Modelo jerárquico**: una jerarquía de fichas, cada ficha puede contener otras listas o fichas. Se dan redundancias y puede dar lugar a inconsistencias en la Base de Datos.
- **Modelo en red**: Añade elementos más complejos para evitar los problemas de duplicidad de datos y posibles inconsistencias.
- **Modelo relacional**: Se representan las bases de datos en forma de tablas que se relacionan unas con otras.

SGBD Relacionales: MySQL, SQL, Access, Oracle, Interbase, Paradox, Derby.

Bases de datos orientadas a objetos, el modelado de la realidad se hace más abstractamente para separar el concepto sobre el que se basa el planteamiento de la estructura de la base de datos, de la realizar física que la determina. Se rigen por las mismas especificaciones que la programación orientada a objetos. Modelar unidades que interactúen entre sí teniendo en cuenta únicamente su aspecto exterior, sus posibilidades de flujo de datos, en lugar del método con que hayan sido codificadas. Almacenan directamente objetos.

Características:

- Independencia de los datos.
- Seguridad.
- Evitar redundancias.
- Consistencia.
- Facilidad de uso para el usuario.

Recomendaciones:

- Posibilidad de tratar objetos complejos.
- Encapsulamiento.
- Tipos y clases.
- Herencias.

La característica más importante es la independencia de los datos y la consistencia, así como evitar redundancias es lo que se llama concurrencia y recuperación.

SGBD orientada a objetos: Visual FoxPro, PostgreSQL.

5.2. Hojas de Cálculo

Nos permitirán trabajar con fórmulas, cuentas y gestionar datos numéricos. Es una cuadrícula en la que se permiten relacionar las diferentes celdas mediante fórmulas, enlaces, etc. Para realizar operaciones de cálculo, ordenaciones, enlaces a datos, etc.

6. Tratamiento de textos

Nos permiten crear, editar, modificar e imprimir documentos.

Incluidos en los paquetes ofimáticos como en OpenOffice o LibreOffice con el Writer y el Microsoft Office con el Word.

BIBLIOGRAFÍA

Cuerpo General Auxiliar de la Comunidad Autónoma de Cantabria.
Temario Materias Comunes Volumen 2.
Convocatoria 2019-2020.
Editorial Mad.
Sie7e Editores.
Noviembre 2019.
Autores: José Manuel González Rabanal (Licenciado en Derecho), Miguel Ángel Navas Dueñas (Ingeniero Superior en Telecomunicaciones), Sergio Jimeno Molins (Ingeniero Superior en Telecomunicaciones).