

Tema 4: Organización Indizada

- **Introducción.**
- **Ficheros Indizados (organizaciones indizadas)**
- **Índices Multinivel**
- **Indización por Árboles**
- **Indización Mediante Claves Secundarias**



@LABDA, Univ. Carlos III

Tema 4.1: Introducción

Las organizaciones base suelen establecerse entorno a un proceso privilegiado (o unos pocos procesos).

Serial → privilegia inserciones

Secuencial → privilegia acceso ordenado y localización CO

Direccionada → privilegia localizaciones a través de CD

Si queremos acceder por otras claves de búsqueda. ¿qué necesitamos? Realizar una búsqueda serial....

Motivación para la indización: almacenar en un fichero aparte una relación entre la clave de búsqueda y su localización en el fichero.



@LABDA, Univ. Carlos III

Tema 4.1: Introducción. Definición

ÍNDICE: directorio cuya entrada se refiere a un solo registro

- Es como un listado para traducir punteros (lógicos a relativos)

- “Almacenamiento auxiliar utilizado para localizar los registros”
 - Los índices se almacenan en un fichero (o varios) de índices.
 - Los registros se almacenan en un fichero de datos o *fichero principal*
 - El fichero principal mantendrá su *organización básica*
- El acceso selectivo a los registros a través del índice se hará mediante una clave que recibe el nombre de *clave de indización*.
- Tipos de índice: según sea la clave de indización
 - si es clave de identificación → **índice primario**
 - cualquier clave de búsqueda no unívoca → **índice secundario**



@LABDA. Univ. Carlos III

Tema 4.1: Introducción. Propiedades

- Un fichero puede contar con más de un índice (*índices múltiples*)
Posibilidad de buscar por distintas claves.

Asi-Fin	3	•	Dum-Mos	Los Tres Mosqueteros	Dumas	...
Dum-Con	4	•	Per-Cap	El Capitán Alatriste	Pérez-Reverte	...
Dum-Mos	1	•	Asi-Fin	El Fin de la Eternidad	Asimov	...
Per-Cap	2	•	Dum-Con	El Conde de Montecristo	Dumas	...
...

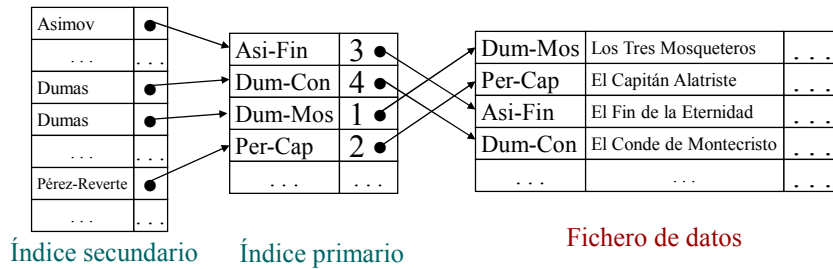
Asimov	•	Dum-Mos	Los Tres Mosqueteros	Dumas	...
Dumas	•	Per-Cap	El Capitán Alatriste	Pérez-Reverte	...
Dumas	•	Asi-Fin	El Fin de la Eternidad	Asimov	...
Pérez-Reverte	•	Dum-Con	El Conde de Montecristo	Dumas	...
...



@LABDA. Univ. Carlos III

Tema 4.1: Introducción. Propiedades

- Es deseable que el índice esté almacenado en memoria principal (total o parcialmente) para optimizar el acceso (selectivo).
- Si se cuenta con más de un índice, se puede considerar un índice primario (poco volátil) y referir el resto a este.



Es conocido como **CLUSTERING**



@LABDA, Univ. Carlos III

Tema 4.1: Introducción. Ventajas

¿ Por qué un índice ?

(1) Aumento de la Tasa de Acierto

- **Hit:** Coincidencia en una búsqueda
- **Hit Rate:** Tasa de acierto (coincidencia) de una búsqueda sobre un soporte

Ejemplo: fichero secuencial; 10^5 registros de 1Kb; 1024 Kb de $M_{intermedia}$
una dirección de absoluta ocupa 2 caracteres, la clave 1 carácter

- Sin índice:

en M_{int} tenemos 1024 reg. → Hit Rate $\approx 1\%$ (casi siempre hay que buscar)

- Con índice

caben $1024^2/3$ bytes = 349525 entradas de índice → Hit Rate (índice) = 100%

- El índice no nos da el registro, pero sí su dirección (localización inmediata)
- El 100% de las veces, hacemos 1 acceso



@LABDA, Univ. Carlos III

Tema 4.1: Introducción. Ventajas

(2) Acceso por claves alternativas

Ejemplo 1: fichero secuencial → con clave de ordenación

Ejemplo 2: fichero direccionado → con clave de direccionamiento

- si la clave de búsqueda es otra → la búsqueda es serial
- si el hit rate es bajo, habrá que leer todo el fichero casi siempre

Solución 1: índice que relacione *clave_búsqueda* → *clave_ordenación* /*direccionamiento* hará que la búsqueda sea dicotómica (coste logarítmico) o AT

Solución 2: índice que relacione *clave_búsqueda* → *dirección_física* búsqueda inmediata, pero si los registros cambian de posición (por copia, reorganización, etc.) habría que reorganizar el índice

clave_búsqueda → *dirección_relativa* (cierta independencia)



@LABDA, Univ. Carlos III

Tema 4.1: Introducción. Ventajas

(3) Reorganización más cómoda

Ejemplo: fichero secuencial → con clave de ordenación

- genera un área desordenada al final del fichero
- precisa reorganizaciones periódicas

Solución: puedo tener el fichero desordenado (org. serial) pero con un índice ordenado (clave de ordenación → posición física o relativa)

Ventaja: se sigue precisando reordenar el índice (exactamente igual) pero es mejor reorganizar algo que cabe en memoria ppal.

- Se reorganiza con facilidad
- Actualizar (almacenar) los cambios es poco costoso
- Antes de actualizar los cambios, se aprovecha la reorganización

Inconveniente: ¿qué ocurriría si el índice no cupiera en memoria principal?



@LABDA, Univ. Carlos III

Tema 4.2: Ficheros Indizados. Estructura

- Estructura de una Organización Indizada :
Fichero de Datos + **Fichero de Índices** [+ Fichero de Desbordamientos]
- Parámetros de una Organización Indizada:
 - *el tipo de índice*
 - *la organización del índice*
 - *la organización del fichero de datos*
 - *la gestión de desbordamientos*



@LABDA. Univ. Carlos III

Tema 4.2: Ficheros Indizados. Tipos

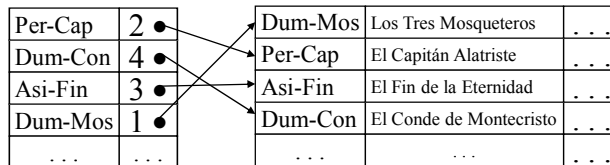
- Según el carácter identificativo de la clave de indización:
 - *si es clave de identificación* → **índice primario**
 - *cualquier clave de búsqueda no unívoca* → **índice secundario**
- Según la correspondencia (biyectiva o no) entre entradas y registros:
 - **Denso** (1:1): existe una entrada del índice para cada registro
 - **No Denso** (1:n): varios registros indizados por una sola entrada
- Según el recubrimiento del índice:
 - **Exhaustivo**: todos los registros que deben tener entrada la tienen
 - **Parcial**: no se indizan todos los registros
(*se dejan aparte los que se acceden rara vez, los últimos en ser introducidos, etc.*)
 - *Si el índice exhaustivo falla, indica que el elemento buscado no existe;*
pero si un índice parcial falla, no aporta ningún valor informativo.



@LABDA. Univ. Carlos III

Tema 4.2: Ficheros Indizados. Tipos

tipo de índice	organización del índice	organización del f. datos
<i>denso</i>	<i>serial</i>	<i>serial</i>



Propiedades:

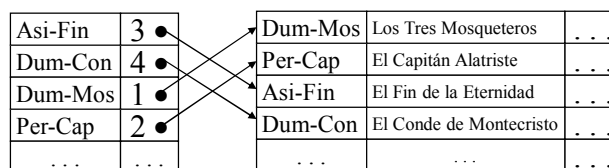
- localización: serial sobre el índice (mucho tiempo de proceso)
si el índice cabe todo en memoria, pocos accesos (1)
- tamaño del índice: gen. grande, ya que cada registro tiene una entrada.



@LABDA, Univ. Carlos III

Tema 4.2: Ficheros Indizados. Tipos

tipo de índice	organización del índice	organización del f. datos
<i>denso</i>	<i>secuencial</i>	<i>serial</i>



Propiedades

- localización: ordenada sobre el índice (menor t. de proceso)
- el resto de consideraciones, no cambian
(salvo que en este caso aparece la necesidad de reorganizar)



@LABDA, Univ. Carlos III

Tema 4.2: Ficheros Indizados. Tipos

tipo de índice	organización del índice	organización del f. datos
<i>denso</i>	<i>serial</i>	<i>secuencial</i>
	<i>secuencial</i>	

Ventajas:

- (1) si $CI = CO$; posibilidad de utilizar una técnica para procesos a la totalidad, y otra para procesos selectivos.
- (2) si $CI \neq CO$ aporta variedad en las claves de búsqueda.

Desventajas:

- (1) si $CI = CO$; reorganizaciones tanto en el fichero como en el índice.



@LABDA, Univ. Carlos III

Tema 4.2: Ficheros Indizados. Tipos

tipo de índice	organización del índice	organización del f. datos
<i>denso</i>	<i>serial</i>	<i>direccionado</i>
	<i>direccionado</i>	

Propiedades

Es obligado que la $CI \neq CD$. Permite buscar por claves de búsquedas distintas de la CD

tipo de índice	organización del índice	organización base
<i>denso</i>	<i>secuencial</i>	<i>direccionado</i>

Propiedades

Si la $CI = CD$ se mejora la consulta ordenada a la totalidad

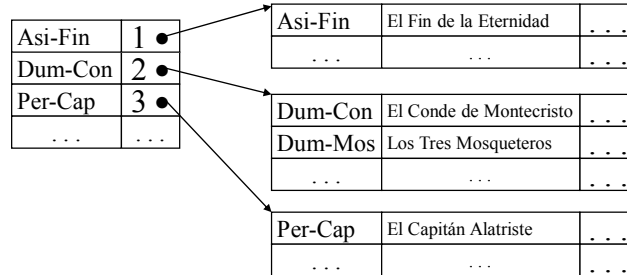
En otro caso estaríamos en el caso anterior aunque con una ordenación lógica del fichero direccionado



@LABDA, Univ. Carlos III

Tema 4.2: Ficheros Indizados. Tipos

tipo de índice	organización del índice	organización del f. datos
<i>No denso</i>	<i>secuencial</i>	<i>secuencial</i>



- localización:
 - búsqueda ordenada en el índice → se obtiene dirección del cubo
 - se recupera el cubo, y se busca dentro del cubo ordenado



@LABDA, Univ. Carlos III

Tema 4.2: Ficheros Indizados. Tipos

Propiedades de los Índices No Densos:

- Sólo puede aplicarse si la organización base es secuencial (ordenada), el índice es secuencial (ordenado), y *clave_indización = clave_ordenación*
- *sólo puede existir un índice no denso para cada fichero de datos*
- Aporta varias posibilidades de acceso:
 - 1.- procesos ordenados (a la totalidad): acceso serial (ordenado)
 - 2.- procesos selectivos (solución única): a través del índice
 - 3.- mixtos (selección de un rango): acc. indizado (1er elemento) + serial



@LABDA, Univ. Carlos III

Tema 4.2: Ficheros Indizados.

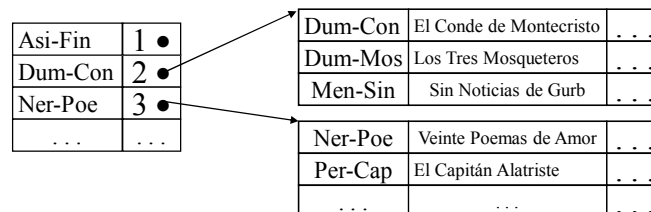
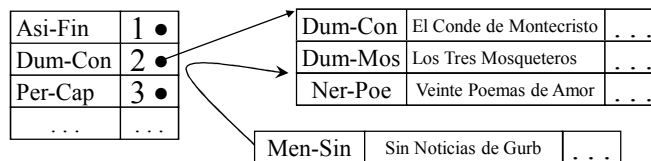
Propiedades de los Índices No Densos:

- Surge el clásico problema de desbordamientos, con sus consecuencias:
 - 1.- Gestión por Área de Desbordamiento (con o sin encadenamiento)
A veces se considera en un fichero aparte (fichero desbordamientos)
 - 2.- Reorganización:
¡la reorganización puede provocar cambios en el índice!



@LABDA, Univ. Carlos III

Tema 4.2: Ficheros Indizados. Tipos



@LABDA, Univ. Carlos III

Tema 4.2: Ficheros Indizados. Interacción

- **Operaciones sobre Ficheros Indizados:**

- Operaciones sobre la totalidad del fichero:

Creación: hay que crear el índice (al crear el fichero o posteriormente)

Borrado: si se borra el fichero de datos, hay que borrar el índice
- se puede destruir el índice sin borrar el fichero de datos.

Consulta: generalmente, interesa hacerla sin contar con el índice
(excepción: proceso ordenado, con índice ordenado y f. de datos serial)

- Operaciones selectivas (sobre registros aislados):

Localización: se hará a través del índice

Operaciones de Actualización: pueden afectar al índice
(es necesario actualizarlo)



@LABDA. Univ. Carlos III

Tema 4.2: Ficheros Indizados. Interacción

- **Operaciones Selectivas sobre Ficheros Indizados:**

- Recuperación:

Consulta Selectiva: localización + recuperación del registro
(si el índice es *no denso*, se recupera un cubo y se busca dentro el registro)

- Actualización:

Inserción:

- índice denso: inserción en ambos (f. de datos e índice)
cada fichero (datos e índice) mantiene sus peculiaridades (orden, ...)

- índice no denso: inserción sólo en fichero de datos
(puede producir desbordamientos, que a su vez pueden afectar al índice)



@LABDA. Univ. Carlos III

Tema 4.2: Ficheros Indizados. Interacción

- **Operaciones Selectivas sobre Ficheros Indizados:**

- Actualización: (cont.)

Borrado:

- índice denso: borrado en ambos → aparecen huecos

Los huecos pueden ser gestionados mediante una *lista de huecos*

- si el fichero de datos es secuencial, la lista de huecos será ordenada
- si no lo es, la lista de huecos podría estar ordenada por tamaños
- si los registros son de longitud fija, simplemente es una lista

- índice no denso: el borrado genera espacio en el cubo (*no pasa nada*)

El problema puede venir si el elemento borrado es el índice de su cubo

En este caso, se hace necesario reorganizar ese cubo y actualizar el índice



@LABDA. Univ. Carlos III

Tema 4.2: Ficheros Indizados. Interacción

- **Operaciones Selectivas sobre Ficheros Indizados:**

- Actualización: (cont.)

Modificación:

- índice denso: modificación en el fichero de datos

- si cambia la clave, modificación en ambos
- si el fichero de datos tiene registros variables, o es secuencial, puede dar problemas: una solución es borrar y reinsertar

- índice no denso: modificación en el fichero de datos

- si cambia la clave y esta es el índice del cubo, será necesario modificar en ambos y comprobar si es necesaria una reorganización
- también pueden dar problemas los registros variables (raro)



@LABDA. Univ. Carlos III

Tema 4.2: Ficheros Indizados.

- **Gestión de Desbordamientos: (solo índices no -densos)**

- Área (o Fichero) de Desbordamiento

- con organización serial
- con encadenamiento
- con punteros múltiples (el índice tiene una colección de punteros)
- nueva entrada en el índice para cada cubo de desbordamiento

- Espacio Libre Distribuido: procurar espacio libre en el área de datos

- intercalar cubos completamente vacíos
(se podría emplear también para secuencial indizado denso, pero no se hace)
- dejar espacio en cada cubo: admite técnicas para mantener ese espacio
 - *rotaciones*: traspasar elementos de un cubo lleno a otro vecino con sitio
 - *particionamiento celular*: dividir un cubo lleno en dos semi-llenos



@LABDA. Univ. Carlos III

Tema 4.2: Ficheros Indizados.

Índices muy grandes

- El **índice denso** puede ser muy grande → elevado coste de almacenamiento y de reorganización (si además es ordenado).
- Un **índice no denso** precisa menos entradas. Así, es apropiado cuando hay muchas entradas y el índice no cabe en memoria
- Si sigue sin caber, habrá que fraccionar el índice.
¿Cómo organizar esos fragmentos de índice?
¿Qué tal con un '*índice de índices*'? → **índice binivel**

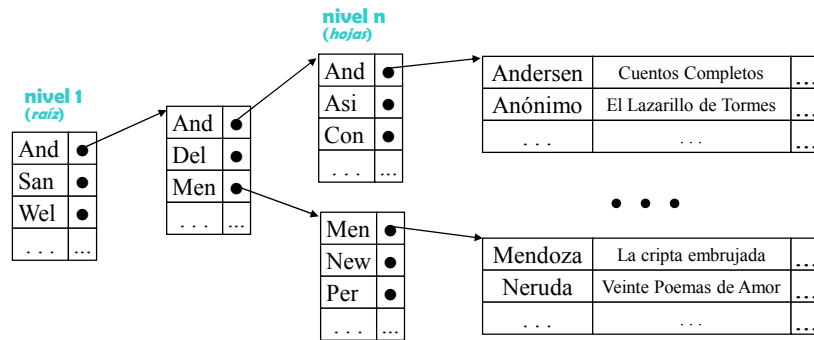
¿Y si ese índice sigue siendo grande? → **índice multinivel**



@LABDA. Univ. Carlos III

Tema 4.3: Índices Multinivel

Son índices con n niveles (el nivel n es índice del nivel $n+1$)



- El número de **niveles es fijo**
- Siempre $n^\circ_{entradas_nivel_n} < n^\circ_{entradas_nivel_n+1}$



@LABDA, Univ. Carlos III

Tema 4.3: Índices Multinivel

Tiempo de Acceso:

- Con un índice de este tipo, es ventajoso que todo el nivel 1 quepa en memoria (ahorramos un acceso al soporte).
- Si un cubo de cualquier nivel se lee en un acceso ($T_{cubo} = T_{bloque}$), el número de accesos para conseguir la dirección de los datos será $n-1$
- Si el cubo de datos se lee en un acceso, el total serán n accesos

Problemas:

- el índice puede crecer y, si no aumenta el cubo, acabará desbordando
- el número de accesos puede hacerse muy elevado
- mantener el índice ordenado es muy costoso (mucha reorganización)



@LABDA, Univ. Carlos III

Tema 4.4: Indización en Árbol

Es un caso particular de índice multinivel (en forma de árbol)

Definiciones:

- Un **árbol** está formado por nodos. Los nodos tienen un tamaño fijo (en cada nodo cabe un número fijo de entradas de índice)
- Cada nodo tiene un solo **padre** (menos la raíz, nivel 1, que no tiene)
- Cada nodo tiene uno o más **hijos** (menos las hojas, nivel n, que no tienen)
- Nodos internos son todos aquellos que no sean ni raíz ni hojas.
- Respecto de un nodo, los nodos que están por encima suyo (de menor nivel) son ascendentes, y los que están debajo (de mayor nivel) son descendientes.
- Un **subárbol** consta de un nodo interno con todos sus descendientes.



@LABDA. Univ. Carlos III

Tema 4.4: Indización en Árbol

Distintas aproximaciones clásicas

- **Árbol binario**. Cada nodo tiene una entrada (clave+puntero al f. de datos) y dos punteros dirigidos a dos nodos hijos (exceptuando las hojas, claro). El recorrido es inorden, es decir, {subárbol_izq} nodo {subárbol_derecho}
Mejora: **Árbol binario paginado** ($T_{\text{pagina}} = T_{\text{bloque}}$)
- **Árbol AVL**. Evita la degeneración de los árboles binarios; por lo tanto mejora el acceso.
Mejora: **Árbol AVL paginado** ($T_{\text{pagina}} = T_{\text{bloque}}$)

acceso med. binario: $\log_2(n+1)$

acceso med. binario paginado: $\log_{m+1}(n+1)$ (k = claves/por página)

acceso max. AVL: $1.44 \cdot \log_2(n+2)$

El gran problema es que no se asegura una ocupación mínima en cada página y las reorganizaciones son muy costosas.



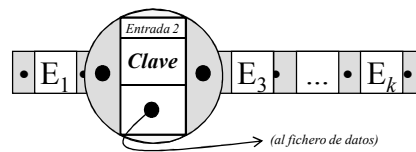
@LABDA. Univ. Carlos III

Tema 4.4: Indización en Árbol B

Árboles B (Bayer/McCreight)

Idea: ya que inicialmente no se conoce el elemento que es mejor separador, se comienza por las hojas. A medida que crezca, se construye hacia arriba.

Nodo: cada nodo va a llenar un bloque; contiene entradas de índice (pares *clave indización-puntero a los datos*) y punteros (para apuntar nodos hijo)



- **Orden del árbol (m)**: indica la capacidad de los nodos
 - según las entradas: el n° mínimo de claves de un nodo (*Bayer*)
 - según los punteros (hijos): n° máximo de hijos de un nodo (*Knuth*)

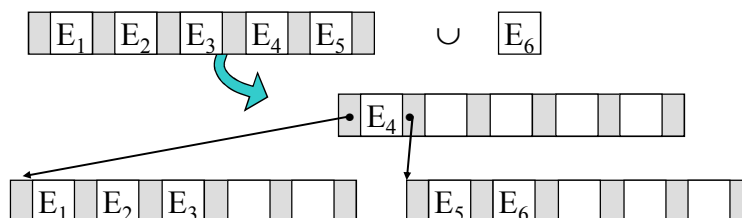


@L.A.B.D.A., Univ. Carlos III

Tema 4.4: Indización en Árbol B

Árboles B : Partición y Promoción

- Las entradas dentro de un nodo van ordenadas
- Cuando un nodo *desborda*, se divide en dos y se promociona el elemento intermedio hacia el nivel superior (ese elemento se lleva dos punteros: uno hacia cada hijo, es decir, hacia cada uno de esos dos nuevos nodos)
<http://slady.net/java/bt/> y <http://articulos.conclase.net/arboles-b/>



@L.A.B.D.A., Univ. Carlos III

Tema 4.4: Indización en Árbol B

Árboles B : Observaciones

- Si el árbol es de orden m , cualquier nodo tendrá a lo sumo m descendientes
- Si un nodo tiene m descendientes (no hoja), tendrá $m-1$ entradas
- Corolario: un nodo de un árbol de orden m tiene a lo sumo $k=m-1$ entradas

$$m \cdot T_{\text{puntero}} + k \cdot T_{\text{entrada}} \leq T_{\text{nodo}}$$

- El nodo raíz tiene al menos un elemento y, por tanto, al menos 2 hijos.
- Todas las hojas están al mismo nivel \rightarrow árbol '*plano*' (bushy)
- La inserción de una nueva entrada puede dar lugar a una (o más) particiones
- El borrado de una entrada puede dar lugar a una (o más) concatenaciones



@LABDA, Univ. Carlos III

Tema 4.4: Indización en Árbol B

Árboles B : Propiedades

- Todos los nodos menos el raíz garantizan una ocupación mínima:

$$k_{\min} = \lfloor \frac{k}{2} \rfloor$$

Corolario

- ¿Cuántos descendientes tienen los nodos intermedios?

(suponiendo política de '*dividir cuando desborda*')

$$m_{\min} = k_{\min} + 1$$

$$m_{\min} = \lfloor \frac{m+1}{2} \rfloor$$

- Tamaño del fichero de índices

Se puede obtener una cota superior del fichero de índices

$$N_{\max}^{\text{nodos fichero}} = (n^{\circ} \text{entradas fichero} - 1) / k_{\min} + 1 \text{ (raíz)}$$

$$T_{\max}^{\text{fichero}} = n_{\max}^{\circ} \text{ nodos fichero} \cdot T_{\text{nodo}}$$



@LABDA, Univ. Carlos III

Tema 4.4: Indización en Árbol B

Árboles B : Propiedades (II)

- El nº de niveles (n) para un árbol de orden m y e entradas tiene cota superior:

nivel	nodos	entradas	acumulado
1	1	1	1
2	2	$2 \cdot k_{\min}$	$1 + 2 \cdot k_{\min}$
3	$2 \cdot m_{\min}$	$2 \cdot m_{\min} \cdot k_{\min}$...
...
$\rightarrow n$	$2 \cdot m_{\min}^{n-1}$	$2 \cdot m_{\min}^{n-1} \cdot k_{\min}$	$(2 \cdot m_{\min}^{n-1}) - 1$
$n+1$	$2 \cdot m_{\min}^n$	$2 \cdot m_{\min}^n \cdot k_{\min}$	$(2 \cdot m_{\min}^n) - 1$

Cota Superior:

$$n \leq 1 + \log_{\lfloor \frac{m+1}{2} \rfloor} \left(\frac{e+1}{2} \right)$$

$\leq e$

$> e$ (este nivel ya no vale)



@LABDA, Univ. Carlos III

Tema 4.4: Indización en Árbol B

Árboles B : Propiedades (III)

- El número de accesos al soporte coincide, en general, con el nº de niveles.

Pues en general la raíz estará siempre en memoria

- El tiempo de acceso (máximo) a registro aleatorio dependerá del número de niveles (n) y del tiempo de acceso a un nodo:

$$t_{\max} = n \cdot t_{\text{nodo}}$$

Este tiempo de acceso será cota superior del tiempo de acceso en cualquier momento de la vida del fichero (mientras no cambien sus condiciones).

- Puede calcularse una cota inferior en el tpo. de acceso en base al número de niveles del árbol perfectamente construido, es decir, los nodos con su máxima ocupación.



@LABDA, Univ. Carlos III

Tema 4.4: Indización en Árbol B

Árboles B: Valoración

Aspectos Positivos:

- Existe una cota superior razonable del número de accesos a soporte
- Generalmente, las operaciones (de inserción o borrado) requieren reestructurar un nodo. Y si son más, suelen ser pocas nodos.
- En el peor caso, los nodos están ocupadas a la mitad (aproximadamente)

Aspectos a Mejorar

- Si las entradas son grandes, el orden puede ser demasiado pequeño
- La densidad (mínima) de los nodos es muy mejorable
- En las hojas se desperdicia mucho espacio (no necesitan punteros desc.)

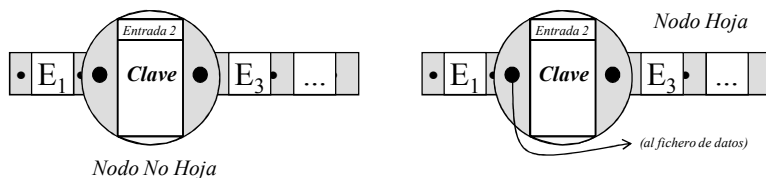


@L.ABDA. Univ. Carlos III

Tema 4.4: Indización en Árbol B+

Idea: se pretende reducir el tamaño de las entradas de índice

Se va a considerar que los nodos intermedios son separadores, y que todas las entradas de índice están en las hojas. En los nodos no hojas sólo será necesaria la clave (se ahorra el espacio; el de el puntero al f. de datos). Por tanto, en cada página caben más claves (el orden es mayor).



Por otro lado, las hojas, al no tener hijos, no utilizan esos punteros. Ahora los van a utilizar para apuntar al área de datos (puntero izqdo. = puntero externo)

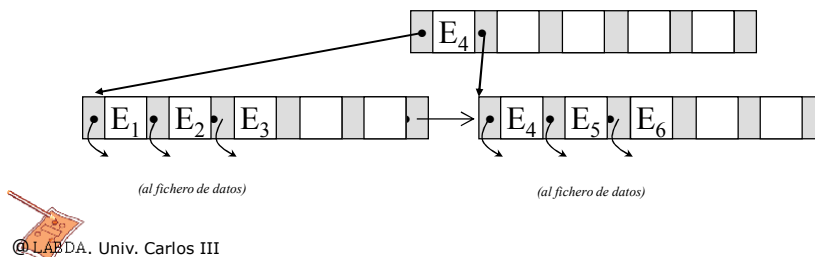


@L.ABDA. Univ. Carlos III

Tema 4.4: Indización en Árbol B+

Árboles B+ : Partición y Promoción

- El funcionamiento es similar al de árboles B. Sólo cambia en la **promoción**: el elemento que promociona desde una hoja, se tiene que **quedar** también en una de las **hojas** resultantes de la división.
- Observar que en los nodos hoja sobra un puntero (el del extremo derecho). Cuando se divide, en el hijo de la izquierda este puntero se asigna apuntando a su hermano (el hijo de la derecha). De este modo, se tienen todas las páginas hoja **encadenadas**.
<http://www.cs.msstate.edu/~cs2314/global/BTreeAnimation/visualization.html>



Tema 4.4: Indización en Árbol B+

- La ocupación mínima de las hojas será:

$$k_{\min} = \lfloor \frac{k+1}{2} \rfloor$$

(suponiendo política de 'dividir cuando desborda')

- La ocupación mínima de los nodos intermedios será $\lfloor k/2 \rfloor \rightarrow$

$$m_{\min} = \lfloor \frac{m+1}{2} \rfloor$$

(se operan como los nodos de un árbol B)

Cálculo del número de niveles:

- El nivel de las hojas es el **nivel n**. ¿Cuántas hojas? $\left\{ \begin{array}{l} n^{\circ} \text{ hojas} = \lfloor e / k_{\min} \rfloor \\ e = n^{\circ} \text{ total de entradas} \end{array} \right.$

- El n° de nodos en el **nivel n-1** depende del número de nodos del nivel n

$$n^{\circ} \text{ nodos } (n-1) = \lfloor n^{\circ} \text{ nodos } (n) / m_{\min} \rfloor$$

- Cuando se llega a un nivel con un solo nodo (la raíz), este será el **nivel 1**.

Tema 4.4: Indización en Árbol B+

Árboles B+ : Observaciones

- El **orden es mayor** que el de los árboles B. Esto hace que existan menos nodos separadores. En consecuencia, la profundidad del árbol será menor (menor o igual en general) ¡aunque se repitan entradas!
- Todos los elementos tienen el **mismo coste de acceso**. Aunque algunos elementos empeoren, la media del coste de acceso mejora.
- El encadenamiento de las hojas proporciona un **método de acceso alternativo** (es un árbol que proporciona acceso secuencial y selectivo)
- El tamaño del fichero de índices es el número total de nodos (la suma de los nodos necesarios entre todos los niveles) por el tamaño de un nodo.



@LABDA. Univ. Carlos III

Tema 4.4: Indización en Árbol B*

Idea: se pretende **aumentar la densidad** de los nodos

Para ello, en lugar de dividir un nodo en dos, se dividirán dos nodos en tres. Así, en lugar de conseguir una ocupación mínima del 50% se obtendrá el 66%

- Cuando un nodo desborda, en lugar de dividir, se procurará ceder uno de sus elementos a su vecino (rotación). Si su vecino también está lleno, se dividen.
- Aparece otra ventaja: *un desbordamiento no siempre supone división/promoción*
- Por lo demás, el resto del funcionamiento es como el de los árboles B.

Propiedades: las mismas del **B**, pero teniendo en cuenta que...

- Todos los nodos menos el raíz garantizan una ocupación mínima: $\lfloor \frac{2k}{3} \rfloor$

- Los nodos intermedios cuentan con $\frac{2k}{3} + 1$ descendientes $\rightarrow \lfloor \frac{2m+1}{3} \rfloor$
(suponiendo política de 'dividir cuando desborda')



@LABDA. Univ. Carlos III

Tema 4.4: Indización en Árbol B*

Consideraciones Finales

- Se puede organizar un fichero de datos en árbol (equivale a sustituir el puntero por el correspondiente registro de datos). Sólo interesa si el registro es pequeño y el nodo es grande (el orden ha de ser grande).
- Se pueden combinar las mejoras logradas por árboles B+ y B*
- En general, se buscan variantes que proporcionen mayor eficiencia
 - menor número de accesos
 - menor profundidad del árbol
 - mayor aprovechamiento de espacio
 - mayor orden
 - mayor ocupación
 - mínimo coste de actualización



@LABDA. Univ. Carlos III

Tema 4.5: Indización. Claves Secundarias

- **Índice Secundario:** la diferencia es que la identificación no es unívoca (*pueden existir varias entradas de índice para cada valor de la clave*)
 - Probablemente, el fichero no esté ordenado por la clave secundaria. En consecuencia, interesa contar con un índice denso ordenado.
 - Si se quiere considerar un *índice multinivel*, por el mismo motivo el nivel n del índice tendrá que ser denso
- **Tipos de Índice Secundario:** (atendiendo a la completitud)
 - *Índice Exhaustivo:* todos los registros indizados
 - *Índice Parcial:* sólo se indizan los registros más interesantes



@LABDA. Univ. Carlos III