UNIDAD 1: PLANIFICACIÓN DEL ALMACENAMIENTO E INDEXACIÓN

Ejercicio 1 (Importante: Tamaño de archivos, índice secuencial, índice multinivel, índice árbol B+)

Se dispone de un disco con bloques de B = 512 bytes. Un puntero a un bloque tiene P = 6 bytes de longitud y un puntero a un registro tiene P_r = 7 bytes de longitud. Un fichero tiene 30.000 registros de longitud fija de una tabla EMPLEADO. Cada registro contiene los siguientes campos:

- Nombre: 30 bytes
- NSS: 9 bytes, es la Primary Key de la tabla.
- Código departamento: 9 bytes
- Dirección: 40 bytes
- Teléfono: 9 bytes
- Fecha de nacimiento: 8 bytes
- Sexo: 1 byte
- Código puesto: 4 bytes
- Salario: 4 bytes
- Se utiliza 1 byte adicional como marca de eliminación de registro.

Se pide:

1. Calcular el número de bloques del archivo.
2. Suponer que el fichero está ordenado según el campo NSS y que se desea construir un índice secuencial sobre NSS. Calcular:
   (a) Factor de bloques del índice (Número de registros/bloque).
   (b) Número de entradas y de bloques del índice.
   (c) Si se convierte en un índice multinivel, el número de niveles hasta obtener el índice más eficiente.
   (d) Número total de bloques del índice multinivel
   (e) Número de accesos al bloque si se utiliza el índice primario o el índice multinivel para localizar un NSS en concreto.
3. Suponer que el fichero no está ordenado según el campo NSS y se desea construir un índice secuencial sobre dicho campo. Repetir la sección 2 y comparar los resultados.
4. Suponer que el fichero no está ordenado según el campo NSS y que se desea construir una estructura de acceso de árbol B^+ sobre NSS. Calcular:
   (a) Los órdenes n de los nodos intermedio y nodos hoja del árbol B^+.
   (b) Número de bloques en el nivel de hoja requeridos si los bloques están ocupados aproximadamente al 69% de su capacidad.
   (c) Número de niveles requeridos si los nodos internos están ocupados también al 69%.
   (d) Número total de bloques que ocupa el árbol.
   (e) Número de accesos a bloque para buscar y recuperar un registro del fichero por el campo NSS.

Ejercicio 2 (Importante: Índice árbol B+, índice asociativo secundario campo no clave, índice secuencial sobre campo no clave)

Considerar un archivo de datos que mantiene información sobre estudiantes. Los registros de este archivo tienen los siguientes campos:

Campo	Longitud (bytes)	Observaciones
Carnet	20	Todos los valores diferentes
Nombre	40	
CodCarrera	2	16 carreras diferentes Distribución Uniforme
Edad	16	
IndiceAcademico	32	Valores: 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0 Distribución Uniforme

El archivo de datos tiene 100.000 registros y se desea construir tres índices sobre este archivo:

- El primer índice está definido sobre el atributo Carnet y se implementará usando un árbol B^+.
- El segundo índice se define sobre el atributo CodCarrera y será implementado utilizando un hash estático con la función **CodCarrera mod 8**.
- El tercer índice estará definido sobre el atributo IndiceAcadémico y será un índice secuencial secundario.

Se disponen de bloques de 512 bytes y los punteros a bloques ocupan 6 bytes mientras que los punteros a registros 7 bytes. Además, para el archivo de datos cada bloque se llena al 65% y para los índices, los nodos y los cajones se ocupan al 69%. Se pide:

(a) ¿Cuánto espacio se requiere para almacenar el archivo y sus índices?
(b) Si se desea insertar un registro nuevo de un estudiante. ¿Cuál es el coste de realizar dicha operación?
(c) Si se desea buscar un estudiante por su carnet, ¿Cuánto cuesta esta operación?
(d) ¿Cuál es el coste de listar a todos los estudiantes que cursan una carrera dada? Asumir el peor caso.
(e) ¿Cuál es el coste de listar a todos los estudiantes que tienen un índice académico mayor que 3? Asumir el peor caso.
(f) ¿Cómo se podrían mejorar los procesos de lectura anteriores?

Ejercicio 3 (Importante: Organización asociativa de fichero de datos, índice secuencial primario, índice árbol B+ primario)

Se dispone de 1 fichero de 1 millón de registros, donde cada registro ocupa 200 bytes de longitud de los cuales 10 bytes corresponden al campo clave. Un bloque de disco ocupa 1000 bytes de longitud y un puntero a bloque/registro es de 5 bytes. Se pide:

(a) Usando una organización asociativa del fichero con 1000 cajones, calcular el tamaño del cajón en bloques asumiendo que todos los bloques contienen el

mismo número medio de registros y el tamaño total en bloques del fichero. ¿Cuál es el número medio de accesos necesarios para localizar un registro?

(b) Usando un índice secuencial primario denso de 1 nivel para el atributo clave sobre el fichero y asumiendo que todos los bloques están tan llenos como sea posible, ¿cuántos bloques se necesitan para el índice? Si se emplea una búsqueda binaria sobre el índice, ¿cuantos accesos son requeridos en media para encontrar un registro por su carnet?

(c) Si se usa ahora un árbol B^+ sobre el archivo ordenado y asumiendo que todos los bloques están tan llenos como sea posible, ¿Cuántos bloques se necesitan para el índice? ¿Cuál es la altura del árbol y la principal característica de este árbol?

Ejercicio 4 (Importante: Organización de archivo de datos mediante árbol B+)

Un sistema emplea una organización de la información de imágenes mediante árboles B^+ y páginas/bloques de datos de 4 Kbytes. La información asociada a cada imagen que se quiere organizar ocupa 128 bytes, de los cuales 12 bytes de los anteriores son para el campo de búsqueda. En el servidor se organiza realmente esta información; y la imagen misma se almacena aparte en páginas de datos especiales de 4 MBytes por lo que la información que se organiza ya incluye un enlace a su correspondiente imagen. Sabiendo que en este árbol cada puntero ocupa 4 bytes y que la ocupación media de cada bloque del árbol es del 80%, determinar:

(a) Calcular lo que se comenta a continuación para la organización del archivo:

- Factor de bloque a aplicar a las **páginas/bloques de datos** a organizar, considerando que la información de control de cada página/bloque de datos ocupa 196 bytes.
- Orden del árbol B^+, sabiendo que cada **página/bloque de nodos intermedios/raiz** guarda una información de control de 12 bytes
- Niveles necesarios del árbol para organizar 100.000 imágenes.
- Número máximo de nodos de cada nivel.
- Cantidad de datos que como máximo organiza cada nivel.
- Número máximo de páginas/bloque de datos.
- Número máximo de registros que se podrán almacenar.
- Espacio en Kbytes que ocupa cada nivel como máximo.

(b) El sistema destina como máximo 824 Kbytes para albergar el árbol en memoria principal. ¿Cuántos accesos serán necesarios para localizar y leer una determinada imagen y su respectiva información?

(c) Y si se quisiera obtener todas las imágenes y su respectiva información de forma ordenada según la clave ¿Cuántos accesos a disco se realizarían?

Ejercicio 5 (Importante: Índice en retícula de 1 dimensión)

Considere un índice en retícula (*grid*) sobre un atributo A. Se ha hecho una partición del atributo en 5 rangos, por ejemplo, precios en euros, en los rangos de [0,100), [100,200), [200,300), [300,400), [400,500). Se han indexados 15.000 registros, y cada bloque del índice en retícula puede almacenar claves y punteros hasta 3.300 registros (los registros en sí estarán almacenados en algún otro lugar). Si se tuvieran más registros de un rango, entonces se utilizarían bloques de desbordamiento (*overflow*) de la misma capacidad. Suponga que no existen duplicados del atributo A en los datos.

(a) Suponga que los valores de A están uniformemente distribuidos, esto es, un registro cualquiera puede estar en cualquiera de los cinco rangos con igual probabilidad, y una clave de búsqueda puede estar en cualquier rango con igual probabilidad. ¿Cuál es el número de operaciones de I/O esperado para buscar un registro, dado un valor de A que no está en el índice? *Justifíquelo*.

(b) Ahora suponga que existe un cierto sesgo en los datos. En particular, la probabilidad de que un valor de A (tanto en el registro como en la búsqueda) esté en el rango j es j/15 (tenga en cuenta que la suma de estas probabilidades desde j=1 hasta 5 vale 1). ¿Cuál es el número de operaciones de I/O esperado para buscar un registro, dado un valor de A que no está en el índice? *Justifíquelo*.

(c) Ahora considere un caso de sesgo extremo. La probabilidad de que un valor de A (tanto en el registro como en la búsqueda) esté en el rango 1 es de 1 y, por lo tanto, la probabilidad es 0 para los otros rangos. ¿Cuál es el número de operaciones de I/O esperado para buscar un registro, dado un valor de A que no está en el índice? *Justifíquelo*.

(d) Si se usa un árbol B+ para la misma aplicación, ¿Cuál es el número de operaciones de I/O esperado (nuevamente para un valor que no está en el índice)? Suponga que los nodos de un árbol B+ contienen 3.300 punteros. *Justifique* brevemente porqué.

(e) *Justifique* qué índice, el de retícula o un árbol B+, es mejor para consultas de rangos.

(f) *Justifique* qué índice, el de retícula o un asociativo (*hash*), es mejor para consultas de rangos.

Ejercicio 6 (Importante: índice asociativo)

Considere un índice asociativo sobre un atributo A, y tome como función de hash `h = ⌊ A / 100 ⌋`. Los valores que puede tomar A (precios en €) están en el rango [0, 499.99]. Se han indexados 16.000 registros, y cada bloque de cada cajón puede almacenar claves y punteros para 4.000 registros. Si se tuvieran más registros de un rango, entonces se utilizarían bloques de desbordamiento (*overflow*) de la misma capacidad. Suponga que no existen duplicados del atributo A en los datos.

1. Calcule el número N de cajones distintos necesarios. *Justifíquelo*.
2. Suponga que los valores de A están *uniformemente distribuidos*, esto es, un registro cualquiera puede estar en cualquiera de los N cajones con igual probabilidad, y una clave de búsqueda puede estar en cualquier rango con igual probabilidad. ¿Cuál es el número de operaciones de I/O esperado para buscar un registro? *Justifíquelo*.
3. Ahora suponga que existe un *cierto sesgo* en los datos. En particular, la probabilidad de que un valor de A (tanto en el registro como en la búsqueda) esté en el cajón i es **2*i / N*(N+1)** (tenga en cuenta que la suma de estas probabilidades, desde i=1 hasta N, vale 1). ¿Cuál es el número de operaciones de I/O esperado para buscar un registro? *Justifíquelo*.
4. Ahora considere un caso de *sesgo extremo*. La probabilidad de que un valor de A (tanto en el registro como en la búsqueda) esté en el cajón 1 ó 2 es de 1/2 y,

por lo tanto, la probabilidad es 0 para los otros rangos. ¿Cuál es el número de operaciones de I/O esperado para buscar un registro? *Justifíquelo*.

5. Si se usa un árbol B+ para la misma aplicación, ¿Cuál es el número de operaciones de I/O esperado? Suponga que los nodos de un árbol B+ contienen también 4.000 punteros y claves. *Justifique* porqué.
6. *Justifique* qué índice, el asociativo o un árbol B+, es mejor para consultas de rangos en dos situaciones: la actual y una general.

Ejercicio 7 (Cuestión teórica sobre índices y búsquedas)

Considere una relación R(A,B,C,D,E) que contiene 5.000.000 registros, donde cada página o bloque de la relación almacena 10 registros. R está organizado como un fichero ordenado y además posee índices secundarios. Suponga que A es una clave de R, con valores en el rango 0 a 4.999.999, y que R está almacenada según el orden de A. Se le van a proponer más adelante cuatro expresiones distintas de álgebra relacional. Para cada una de las expresiones, ***justifique*** cual de las siguientes tres situaciones es más probable que sea la mejor (más económica en operaciones de I/O):

- Acceder directamente al fichero de datos ordenado por A
- Usar como índice un árbol-B+ sobre el atributo A
- Usar como índice uno asociativo (*hash*) sobre el atributo A

1. $\sigma_{A < 50{,}000}(R)$
2. $\sigma_{A = 50{,}000}(R)$
3. $\sigma_{A > 50{,}000 \wedge A < 50{,}010}(R)$
4. $\sigma_{A \neq 50{,}000}(R)$