Explicaciones prácticas

Árboles

Árboles B

Son árboles multicamino con una construcción especial de árboles que permiten mantenerlos balanceados a bajo costo. Sus características principales son

- Cada nodo del árbol puede contener como máximo M descendientes y M-1 elementos
- La raíz no posee descendientes directos o tiene al menos 2
- Un nodo con X descendientes directos contienen X-1 elementos
- Todos los nodos (salvo la raíz) tienen como mínimo [M/2] - 1 elementos y como máximo M-1 elementos
- Todos los nodos terminales se encuentran al mismo nivel

Overflow

El overflow se produce cuando un nodo está completo. Cuando esto ocurre debo crear un nuevo nodo, la primera mitad de las claves se mantienen en el nodo con overflow, la segunda mitad de las claves se traslada al nuevo nodo y la menor de las claves de la segunda mitad se promociona al nodo padre.

Bajas

Hay distintos casos

1. Si la clave a eliminar no está en una hoja, se debe reemplazar con la menor clave del subárbol derecho
2. Si el nodo hoja contiene por lo menos el mínimo número de claves, luego de la eliminación, no se requiere ninguna acción adicional
3. Tratar el underflow

Underflow

1. Se intenta **redistribuit** con un hermano adyacente. La redistribución es un proceso mediante el cual se trata de dejar cada nodo lo más equitativamente cargado posible. El elemento más grande del hermano adyacente izquierdo o el más chico del hermano adyacente derecho (según la política) pasa al padre y el elemento divisor del padre pasa al nodo en underflow
2. Si la redistribución no es posible, entonces se debe **fusionar** con el hermano adyacente. Actúa de la siguiente forma: se unen dos nodos hermanos adyacentes más el elemento divisor del padre y forma un solo nodo con la unión de todos los elementos involucrados

Políticas para la resolución de underflow

- Política izquierda: se intenta redistribuir con el hermano adyacente izquierdo, si no es posible, se fusiona con el hermano adyacente izquierdo

- Política derecha: se intenta redistribuir con el hermano adyacente derecho, si no es posible, se fusiona con el hermano adyacente derecho
- Política izquierda o derecha: se intenta redistribuír con el hermano adyacente izquierdo, si no es posible, se intenta con el hermano adyacente izquierdo, si no es posible, se fusiona con el hermano adyacente derecho

Hay un caso especial y es que en cualquier política si se trata de un nodo hoja de un extremo del árbol debe intentarse redistribuit con el hermano adyacente al mismo

SIEMPRE se intentará redistribuir si el hermano adyacente está en condiciones

Árboles B+

Constituyen una mejora sobre los árboles B pues conserva la propiedad de acceso aleatorio rápido y permiten además un recorrido secuencial rápido

- Conjunto índice: proporciona acceso indizado a los registros. Todas las claves se encuentran en las hojas, duplicándose en la raíz y nodos interiores aquellas que resulten necesarias para definir los caminos de búsqueda
- Conjunto secuencia: contiene todos los registros del archivo. Las hojas se vinculan para facilidar el recorrido secuencial rápido. Cuando se lee en órden lógico, lista todos los registros por el orden de la clave

Búsqueda

La operación de búsqueda en árboles B+ es similar a la operación de búsqueda en árboles B. El proceso es simple ya que todas las claves se encuentran en las hojas, deberá continuarse con la búsqueda hasta el último nivel del árbol

Inserción

Es igual que en los árboles B salvo cuando ocurre un overflow. Cuando ocurre, el nodo afectado se divide en 2, distribuyéndose las claves lo más equitativamente posible. Una copia de la clave del medio o la menor de las claves mayores, se promociona al nodo padre. La copia de la clave solo se realiza en un overflow ocurrido a nivel hoja, caso contrario se trata igual que en árboles B

Bajas

La operación de eliminación en árboles B+ es más simple que en árboles B. Ocurre porque las claves a eliminar siempre se encunetran en las páginas hojas. En general deben distinguirse los siguientes casos, dado un árbol B+ de orden M

- Si al eliminar una clave, la cantidad de claves que queda es mayor o igual que [M/2] -1 entonces termina la operación. Las claves de los nodos raíz o internos no se modifican por más que sean una copia de una clave eliminada en las hojas
- Underflow: si al eliminar una clave, la cantidad de llaves es menor a [M/2]-1 entonces debe hacerse una redistribución de claves, tanto en el índice como en las páginas hojas. Si la redistribución no es posible, entonces debe hacerse una fusión entre nodos

---

Hashing

El hashing es una técnica para generar una dirección base única para una clave dada. Convierte la clave en un número aleatorio que luego sirve para determinar dónde se almacena la clave.

Usa una función de dispersión para mapear cada clave con una dirección física de almacenamineto. Es usada cuando se requiere acceso rápido por clave

El hashing puede ser

- Direccionamiento estático: el espacio disponible para dispersar los registros del archivo está fijado previamente
- Direccionamiento dinámico: el espacio disponible para dispersar aumenta o disminuye en función de las necesidades

Parámetros a considerar

- Capacidad de almacenamiento de cara dirección
- Densidad de empaquetamineot: relación entre el espacio disponible para el archivo de datos y la cantidad de registros que integran el mismo. DE= número de registros usados / espacio total
- Función de hash: caja negra que a partir de una clave genera la dirección física donde debe almacenarse el registro
- Método de tratamiento de desbordes: situación en la cual una clave carece de lugar en la dirección asignada por la función de dispersión
- Colisión: situación en la que un registro es asignado, por función de dispersión, a una dirección que ya posee uno o más registros

Aunque la función de dispersión sea eficiente y la densidad de empaquetamiento sea baja, a veces pueden ocurrir desbordes. Los cuales son tratados de la siguiente manera

Satuación progresiva

Cuando se completa una dirección de memoria se busca en las siguientes direcciones en secuencia, hasta encontrar una vacía, su ventaja es la **simplicidad**

- Búsqueda: comienza en la dirección base y continúa buscando en localidades sucesivas hasta encontrar la clave buscada
- Eliminación: no debe permitirse que el espacio liberado por la eliminación obstaculice las búsquedas posteriores, al mimso tiempo debe ser posible usar el espacio liberado para adiciones posteriores. Se marca el espacio liberado con ### (el espacio liberado no rompe la secuencia de búsquedas y el liberado está disponible y puede ser usado en adiciones posteriores) cuando se elimina una clave de una cubeta llena y en la próxima cubeta hay otra marca o dato

Saturación progresiva encadenada

Funciona igual que la progresiva excepto que las claves sinónimos se enlazan por apuntadores cada dirección base contiene un número que indica el lugar del siguiente registro con la misma dirección base. El siguiente registro contiene a la vez un puntero al siguiente registro con la misma dirección base y así sucesivamente.
Su ventaja es que solo se necesita acceder a los registros con claves que son sinónimos por lo tanto mejora el número de accesos promedio

Su desventaja es que debe agregarse un campo de enlace a cada registro y requerirá mayor espacio de almacenamiento

Importante: si la clave en la dirección base a insertar es intrusa, se le asigna nuevo lugar y a partir de la dirección base de la intrusa, se modifica el enlace a su nueva dirección luego se inserta la nueva clave en su dirección base

Satuación progresiva con encadenamiento en áreas separadas

Cuando se agrega un registro nuevo si hay lugar en al dirección base se almacena allí, sino se mueve al archivo de saturación donde se agrega a la lista enlazada que comienza en la dirección base

Importante: si la baja es en el área principal, solamente escribo la cubeta sin la clave a eliminar, no se modifican los enlaces

Dispersión doble

Cuando sucede una colisión se aplica una segunda función de dispersión a la clave, para producir un N.º, el cual se suma a la dirección original tantas veces como sea necesario hasta encontrar una dirección vacía

- Eliminación: se usan marcas ### para futuras búsquedas