

APUNTES CBD

HOJA N°

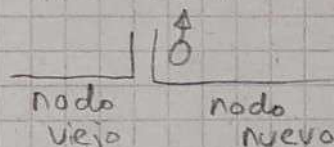
FECHA

Árboles B (orden = cant punteros = cant nodos - 1)

• Alta/Inserción → overflow → división

(se sobrecarga más el
nodo izq)

Y
Promoción



1. E por orden las L

y las E hasta de overflow,

luego los nodos nuevos y luego la raíz o interno
(igual para underflow)

• Baja/Eliminación claves $< ((n/2) - 1)$

Está en nodo hoja con underflow:

- Se elimina y se balancea según política
- Si no se puede balancear fusiona con nodo adyacente

políticas {

- Izg: Se intenta balancear con izq sino derecho
- der: Se intenta balancear con der sino Izquierdo
- Izg & Der: Ante el der underflow se elige una política (Izg & Der) y se continúa resolviendo con esa política
- Izg y Der: Se intenta balancear con alguno, se prioriza el Izquierdo

Está en nodo interno con underflow:

- Se cambia la clave a eliminar con la menor de la rama derecha
- Se elimina la clave de la hoja
- En caso de underflow en la hoja se procede igual política que corresponda

Árboles B+

• Alta/ inserción (se sobrecarga + el nodo derecho)

Está en nodo hoja con overflow:

- Se divide en 2 el nodo afectado y se distribuyen las claves equitativamente
- Una copia de la clave del medio se escribe en el nodo padre
- Se inserta la clave en el nodo derecho (la del medio)

Está en nodo interno con overflow:

- Se procede igual que en árboles B (divide y promociona)

• Baja/ eliminación (L/E → Se escribe zero el otro puntero y luego el puntero del libre y por último la raíz) ⊗ Se escribe zero el que van a fusionar

En nodo hoja

Con underflow → (Cont. de claves $< (n/2) - 1$)

- Se procede igual que en árboles B con la política indicada
- En caso de no poder balancear, se fusiona el nodo con underflow con su hermano adyacente y se pierde el señalador padre

Sin underflow

- Elimino y listo

En nodo interno o raíz

- No se modifican por mas que sean una copia de la clave eliminada
- Se justifica tipo: "No existe la clave en el nodo 4" (nodo hoja)

Hashing

HOJA N°

FECHA

* Hay bits de dispersión de la tabla de direcciones y bits de dispersión para cada bloque

* Si hay 4 registros \rightarrow 4 claves por cubeta

Alta

- No hay overflow \rightarrow Se agrega según la tabla de direcciones y listo

* Si hay overflow

- Se aumenta en 1 el bit de dispersión del bloque con overflow

\rightarrow Si el bit de disp. es menor al de la tabla, NO duplico la tabla:

• Creo un nuevo bloque (con misma cant. de bits de disp.)

• En la dirección con overflow apunto al bloque nuevo y actualizo las direcciones (apuntando también al bloque nuevo) cada 2^n , siendo n los bits de dispersión del bloque

• Redistribuyo fijandome en los últimos bits de las direcciones de las claves

\rightarrow Si el bit de disp. es mayor al de la tabla, duplico la tabla:

• Duplico la tabla de direcciones repitiendo punteros

• En la dirección de overflow apunto al nuevo bloque y luego redistribuyo las claves

• Baja

- Si No hay underflow \rightarrow Se elimina la clave y lista y se reubican a la izquierda las otras claves dentro del bloque

* Si hay underflow:

- Me fijo en su dirección según los bits de dispersión del bloque con underflow y luego me fijo en las direcciones hermanas.

* Si las dire. hermanas son distintas:

- No se libera el bloque
- Se escribe al bloque como vacío

* Si las dire. hermanas son iguales (apuntan al mismo bloque):

- Se libera el bloque (Hay que anotarlo)
- Se sustituye su nro en la tabla por el nro de bloque de las direcciones hermanas. (y se le decrementa en 1 al bloque al que apuntan)

NOTAS

- Las claves van a estar ordenadas de menor a mayor
- direcciones hermanas \rightarrow aquellas que tienen igual último dígito
- Si tengo más de un bloque libre, y se produce un overflow, puedo usar cualquiera pero de preferencia el último en liberarse
- Cant. dire. hermanas = Cant. bits de disp. de la subeta/bloque con underflow
- Si al hacer una alta con overflow tengo un bloque libre, lo utilizo (como en árboles B+B+)