ARBOLES DE BUSQUEDA AVANZADOS

2-3 ARBOLES (Hor & Sahni ED. Pascal Ch. 10.8)

- Nodos con 1 o 2 valores y 2 o 3 hijos
- Todos los hojas a la misma profundidad.

Inserción :
- Simple en 1-hojas
- Partición de 2-hojas y reiteración en el padre
- Posible partición de la raíz con el consecuente crecimiento de la profundidad

Eliminación :
- Simple en 2-hojas
- La eliminación de 1-hojas obliga a reducir al padre y/o a reducir para subir a un hijo. (rotaciones y combinaciones)

2-3-4 ARBOLES (Sedgewick web)

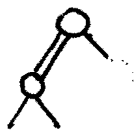
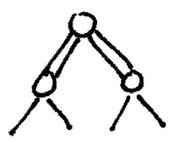
- Nodos con 1, 2 o 3 valores y 2, 3 o 4 hijos.
- Todas las hojas a la misma profundidad.

Inserción :
- En la correspondiente hoja, si se puede.
- Si no cabe, se parte y se manda el intermedio al padre, reiterando el proceso si fuera necesario.
- Variante : Los 4-nodos se parten al bajar
  Ello propaga a la larga el ensanchamiento hacia arriba ; cuando se hace en la raíz la profundidad aumenta.

Implementación farragosa debido a los distintos tipos de nodos

ARBOLES ROJINEGROS (Sedgewick web)

- Implementación de 2-3-4 árboles utilizando ABB's con dos tipos de punteros : negros (ordinarios) y rojos (fabrican 2-3-4 nodos)
- Diversas variantes con distintos tipos de dificultades.
- ARN's sesgados hacia la izquierda ¡NOVEDAD!

3 nodo  4 nodo 

- Color de los punteros almacenado en los nodos apuntados.

Inserción : Mejor recursiva

- Uso de rotaciones sobre punteros rojos inadecuados al insertar rotate Left , rotate Right
- Intercambio de colores para partir un 4-nodo ; posible propagación hacia arriba: combinando intercambios y rotaciones.

- Variante descendente : más sencilla.
  - Las rotaciones se hacen al "subir" (recursión no-final)
  - Código resultante extremadamente sencillo
  - Si hacemos el intercambio de colores al final : ¡ 2-3 árboles gratis !

B^+- árboles (Weiss Ch. 10.7)

- Almacenamiento de registros en bloques de un disco : acceso vía claves.
- Árboles con nodos internos que almacenan claves (entre $M/2$ y M) y hojas que almacenan registros (entre $L/2$ y L)
- Altura logarítmica garantizada ; acceso llegando a las hojas.
- La eliminación se simplifica porque no hay problema en que las claves en los nodos internos no correspondan a elementos del conjunto.
- Mantenemos el invariante de "llenado" de los nodos vía partición en dos al rebasar el máximo y juntando con el vecino al rebasar el mínimo.
- Cuando alcanzamos un caso límite tras repetidas iteraciones del correspondiente proceso, el árbol ve aumentada / disminuida su profundidad.

B - árboles (Cormen Ch. 18)

- Generalizan los 2-3-4 árboles : cada nodo del árbol tiene entre t y 2t hijos, salvo las hojas, que contienen entre t-1 y 2t-1 valores.
- Todas las hojas se encuentran a la misma profundidad.
- Altura logarítmica garantizada.
- Se utiliza la idea de partir nodos llenos al bajar para insertar fácilmente.
- La eliminación exige rotaciones con distinciones de casos bastante pesadas.
