
Estructura de datos y algoritmos

Tema VI

Árboles avanzados

TEMA VI : ÁRBOLES AVANZADOS

- 6.1 Árboles multcamino
 - 6.2 Árboles B
 - 6.3 Árboles B Binarios (árboles BB)
 - 6.4 Árboles B Binarios Simétricos (árboles BBS)
-

6.1 Árboles multcamino

■ **Definición:**

- Se denominan árboles multcamino a aquellos árboles de grado mayor que dos.
 - Árbol que contiene nodos con más de dos ramas.

■ **Aplicaciones:**

- Muy utilizados en la construcción y mantenimiento de árboles de búsqueda con
 - Gran cantidad de nodos
 - Guardados en **memoria secundaria**,
 - En los que **se realizan con frecuencia inserciones y supresiones**.

■ **Idea básica:**

- Un árbol se subdivide en subárboles
- Cada subárbol se representan como unidades a las que se accede simultáneamente y reciben el nombre de *páginas*.
 - Cada acceso a página requiere un único acceso a memoria secundaria.
 - Se aprovecha las características de direccionamiento mediante paginación de la memoria secundaria y se consigue un ahorro en el número de accesos a la misma.

■ **Problemas:**

- (1) ¿De cuántos nodos máximo considero cada página?
- (2) Es necesario establecer un plan de crecimiento controlado del árbol (mínima profundidad) para evitar que este crezca aleatoriamente (podría, en el peor de los casos, haber un nodo por página)

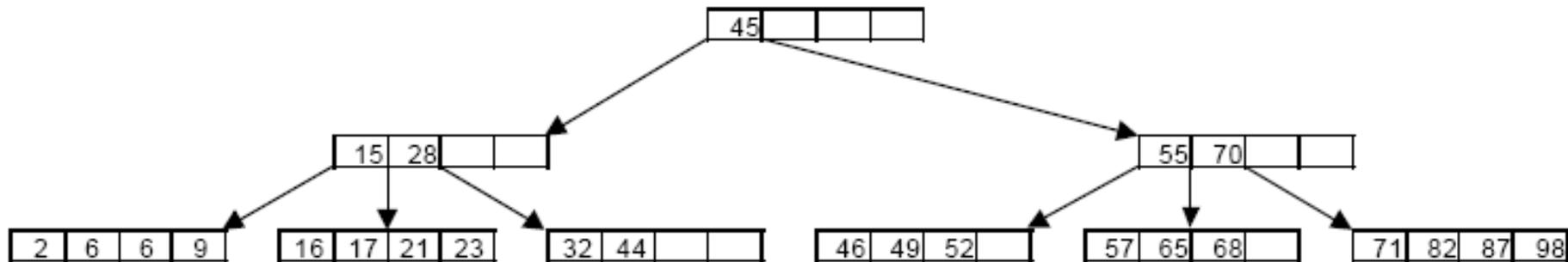
■ **Solución:** Árboles B multcamino o, simplemente, árboles B.

Árboles B

■ Definición:

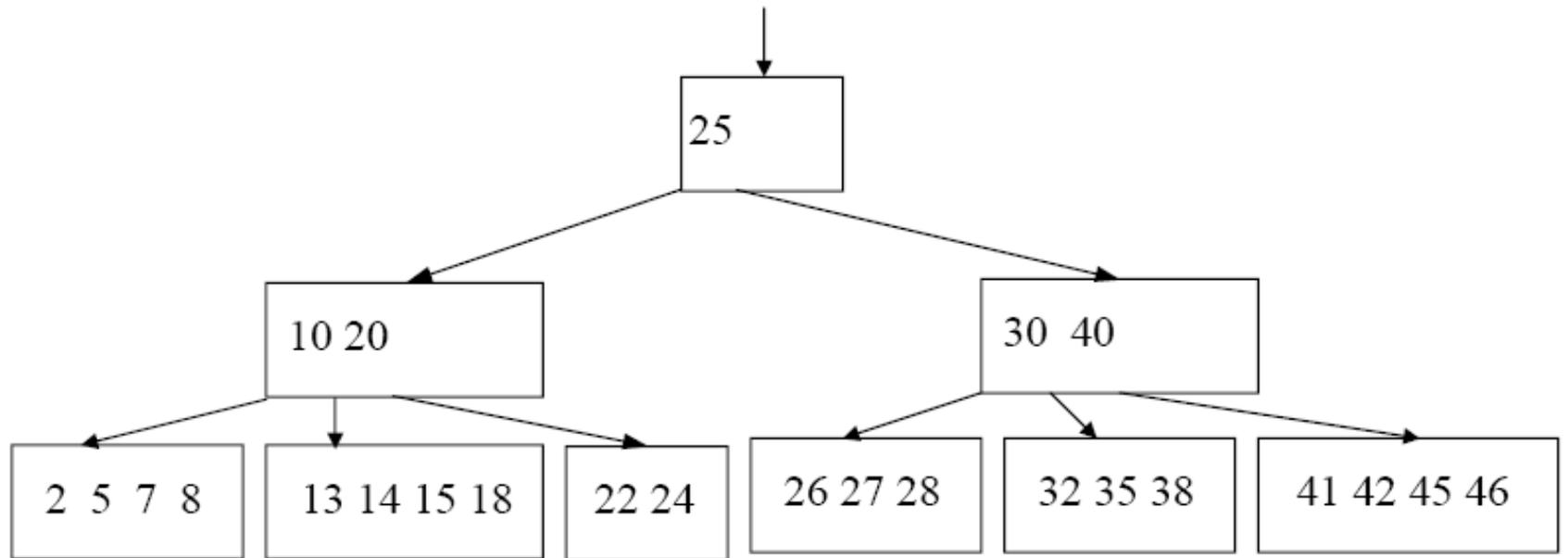
- Un árbol B de **orden n** es aquel árbol de búsqueda que satisface las siguientes propiedades:
 - 1. Cada **página** contiene **máximo $2n$ llaves** y **mínimo n llaves**, excepto la raíz (que puede contener sólo una).
 - 2. Cada página o bien es una página hoja (no tiene descendientes) o bien tiene $m+1$ páginas descendientes, siendo **m** el nº de llaves en esta página.
 - 3. Todas las páginas hoja aparecen al mismo nivel.

Ejemplo de árbol B de orden 2



- *Satisface las condiciones de árbol de búsqueda.*
 - *Cada página contiene como máximo $2 \times 2 = 4$ llaves, y como mínimo 2 llaves, excepto la raíz que almacena una sola.*
 - *Todas las páginas de hoja están en el mismo nivel, el nivel 3, y las que no son de hoja tienen $m+1$ descendientes.*
 - *Así la raíz con una sola llave ($m=1$) apunta a dos páginas, y en el segundo nivel, las páginas contienen dos llaves ($m=2$) y por tanto tienen tres descendientes.*

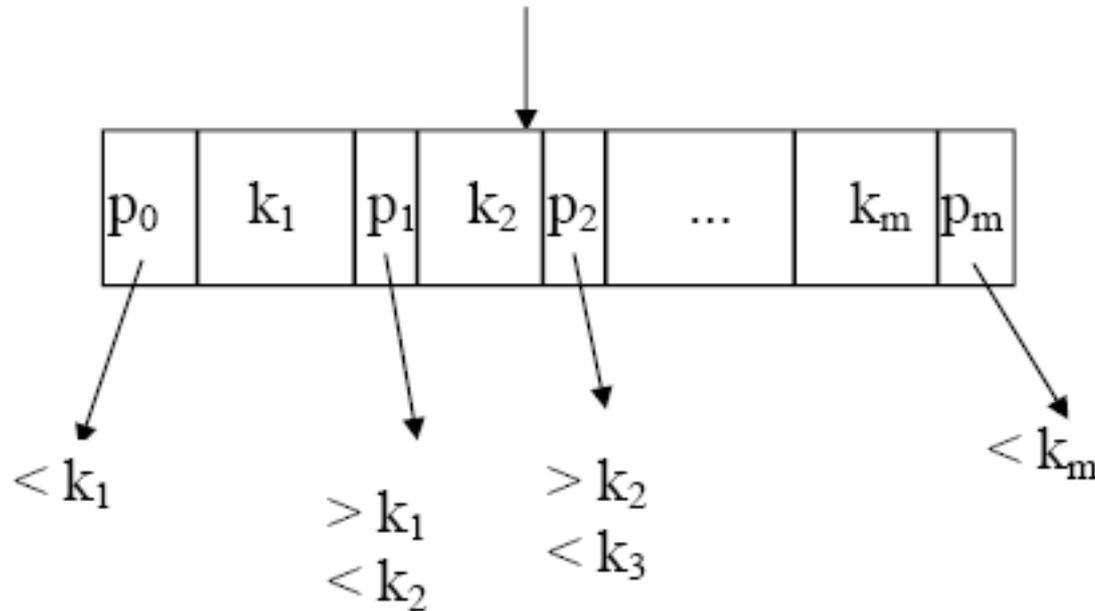
- Ejemplo:



(Arbol B de orden 2 con 3 niveles)

6.2.1 Búsqueda en árboles B

- Representación de una página (**m** llaves y **m+1** punteros)



- Las llaves están ordenadas de izquierda a derecha en cada página y en cada nivel.

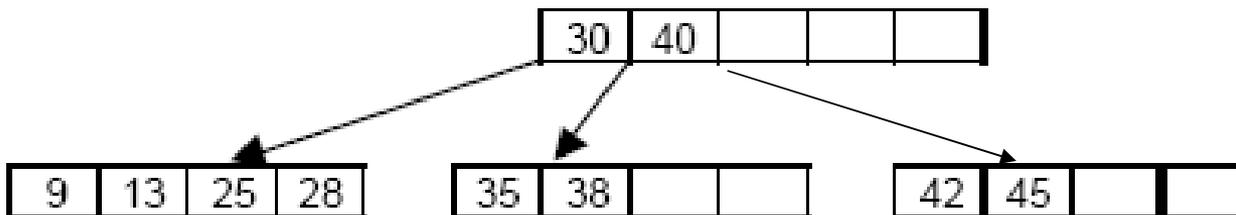
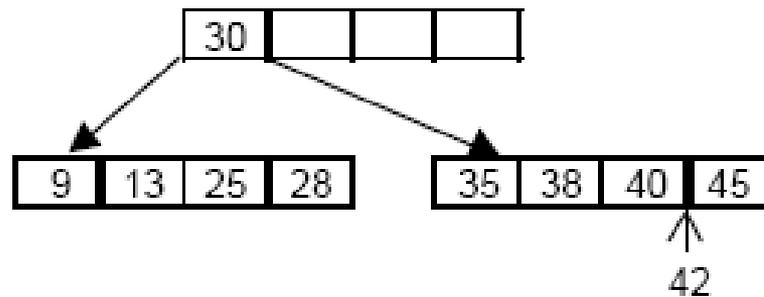
Criterios de búsqueda

- Una vez cargada en memoria principal la página:
 - Si m es grande:
 - Búsqueda binaria
 - Si m es pequeña:
 - Búsqueda secuencial
- Si la búsqueda en esta página fracasa, entonces:
 - 1. Si el argumento de búsqueda se encuentra entre dos llaves de la página, $k_i < x < k_{i+1}$ entonces:
 - La búsqueda sigue en la página apuntada por el puntero entre ellas, p_i
 - 2. Si el argumento de búsqueda es mayor que la última llave de la página, $k_m < x$, entonces:
 - Se sigue la búsqueda en la página apuntada por el último apuntador, p_m
 - 3. Si el argumento de búsqueda es menor que la primera llave de la página, $x < k_1$, entonces:
 - Se sigue la búsqueda en la página apuntada por el primer apuntador, p_0
 - 4. Si en cualquier caso el apuntador a la página en la que hay que seguir la búsqueda es *NIL* entonces:
 - El argumento de búsqueda x no está en el árbol y finaliza la búsqueda.

Inserción en árboles B

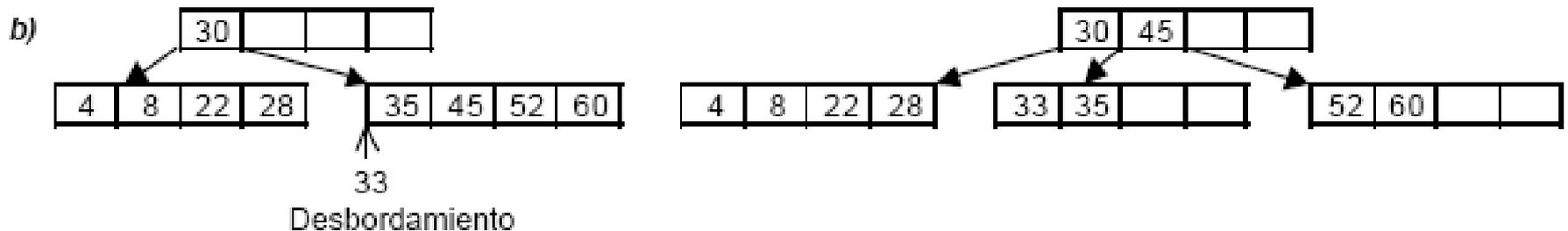
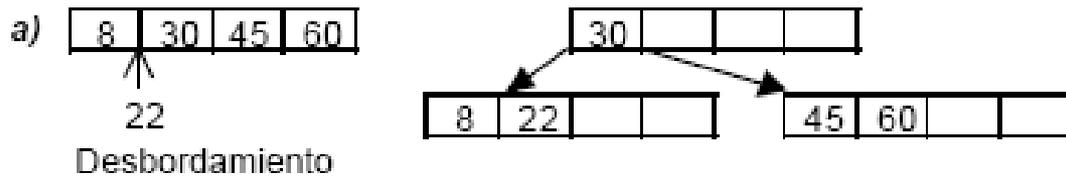
- La inserción se realiza buscando el lugar donde debe insertarse de la misma manera en que se realizó la operación de búsqueda.
- Localizada dicha posición
- Si la página correspondiente tiene $m < 2n$ llaves
 - Se inserta en ella en la posición dada por el criterio de búsqueda.
- Si la página está llena ($m = 2n$) entonces el árbol debe reorganizarse:
 - 1. La página se divide en dos nuevas páginas.
 - 2. La llave que se encontraba en la mitad, ll_{mitad} , se sube un nivel colocándola en la página madre. A la hora de determinar la llave central se tiene en cuenta también la nueva llave que se pretende insertar.
 - 3. Las páginas que se encontraban a la izquierda se distribuyen en la página nueva apuntada por la izquierda de ll_{mitad} y las que se encontraban a la derecha se distribuyen en la página nueva apuntada por la derecha de ll_{mitad} .
- La división de páginas puede propagarse hasta la raíz. Es obvio que el árbol crece de las hojas a la raíz.

Inserción del dato 42 en un árbol B de orden 2



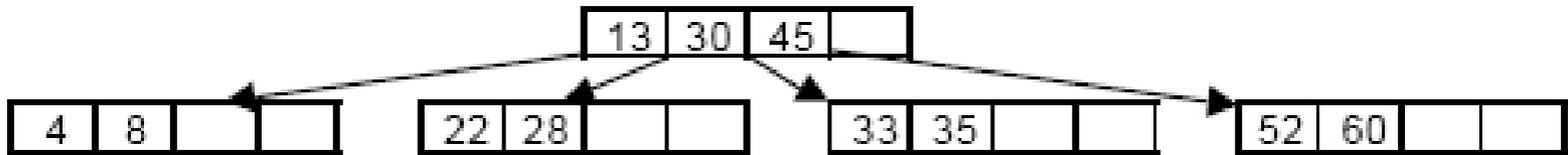
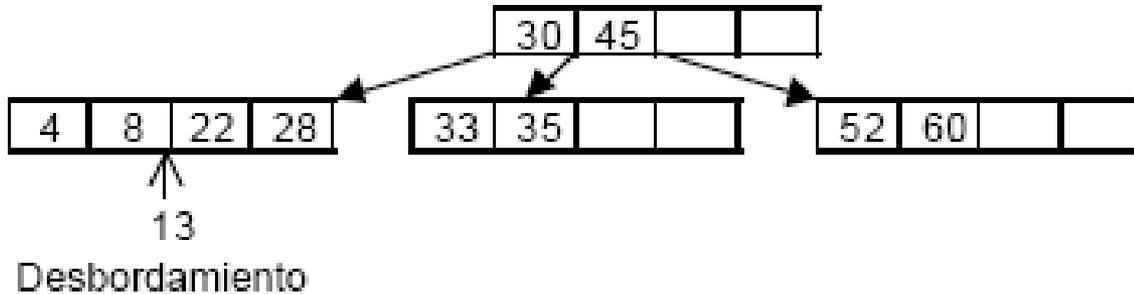
Ejemplo 2

- *Supóngase que en un árbol B de orden dos, inicialmente vacío, se introducen consecutivamente las siguientes llaves:*
 - *30, 60, 45, 8, 22, 35, 4, 28, 52, 33, 13, 39, 41, 43, 24, 25, 15.*



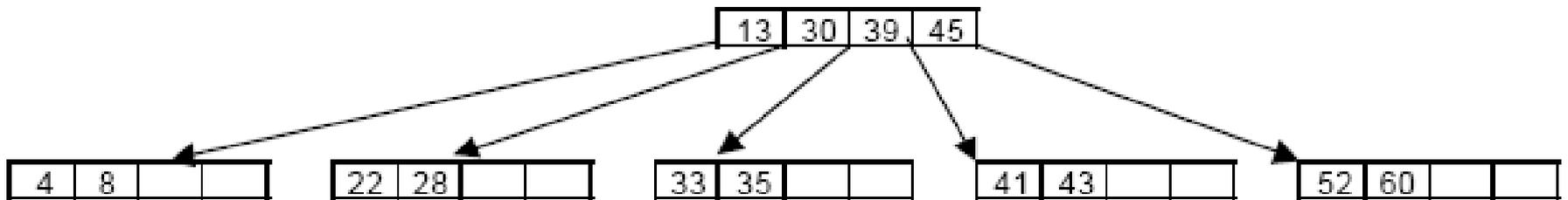
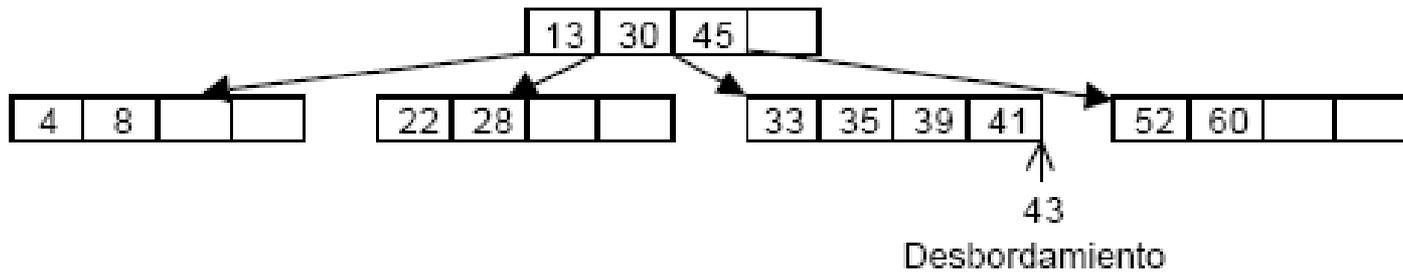
- 30, 60, 45, 8, 22, 35, 4, 28, 52, 33, 13, 39, 41, 43, 24, 25, 15.

c)



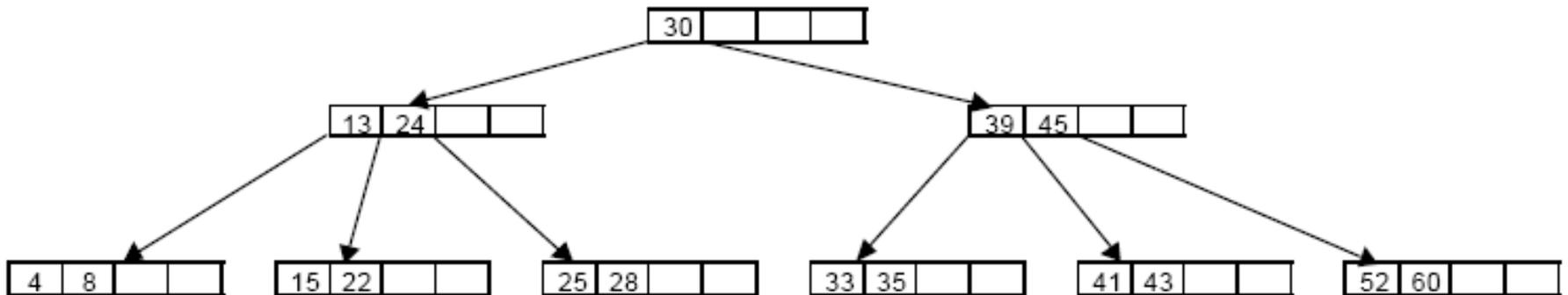
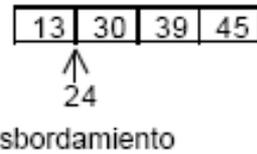
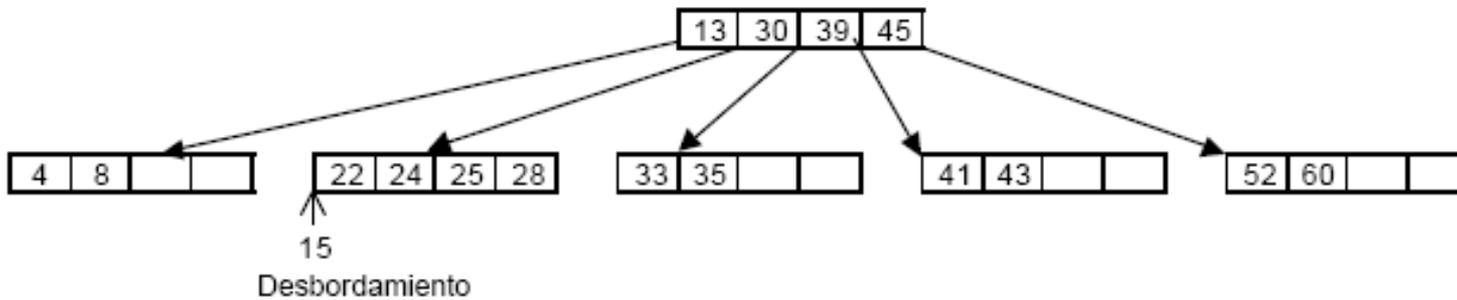
- 30, 60, 45, 8, 22, 35, 4, 28, 52, 33, 13, 39, 41, 43, 24, 25, 15.

d)



- 30, 60, 45, 8, 22, 35, 4, 28, 52, 33, 13, 39, 41, 43, 24, 25, 15.

e)



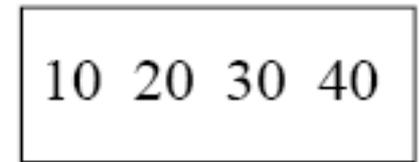
Ejemplo. Construcción de árbol B de orden dos ($n=2$)

- 20; 40 10 30;15;35 7 26 18;22; 5; 42 13 46
27 8; 32;38 24 45; 25

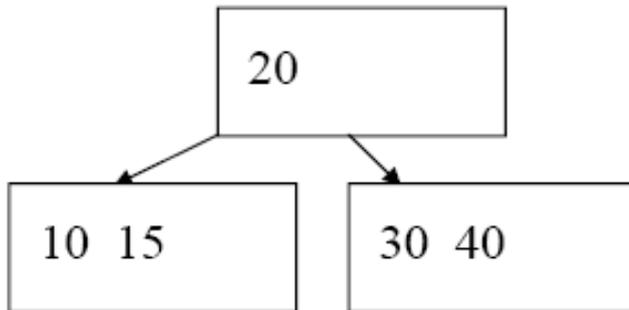
(a) 20



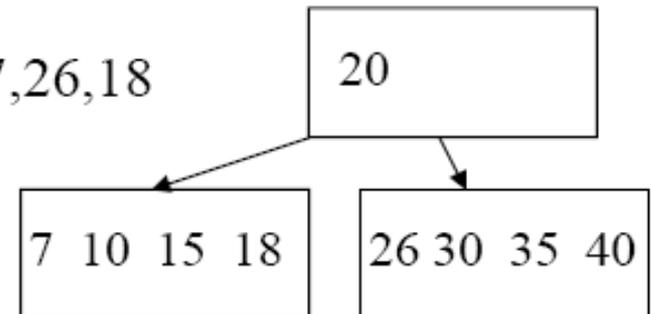
(b) 40, 10, 30



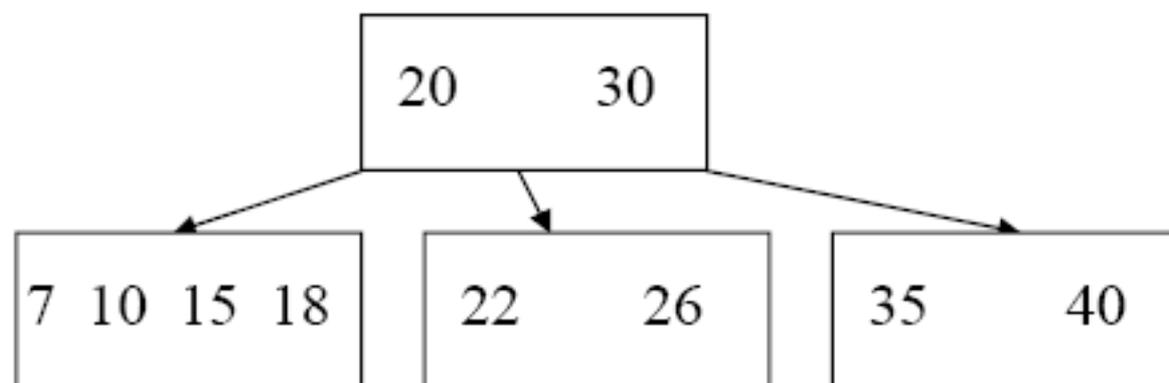
(c) 15



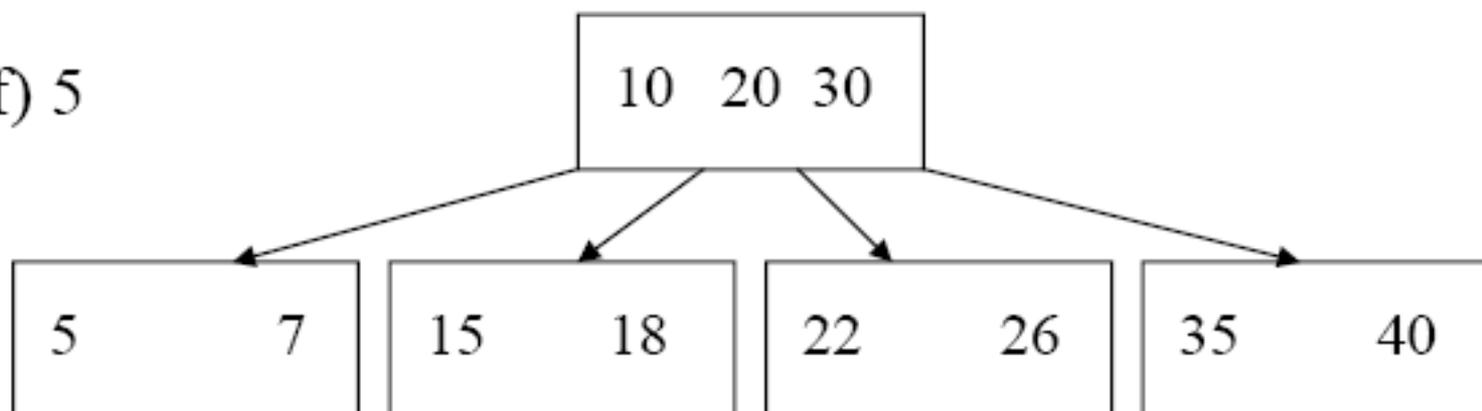
(d) 35,7,26,18



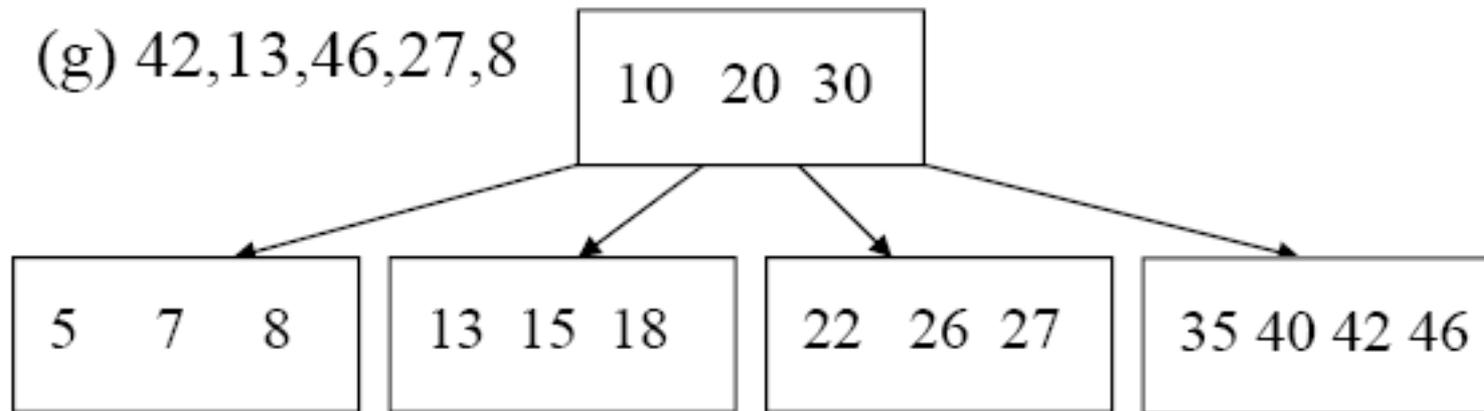
(e) 22



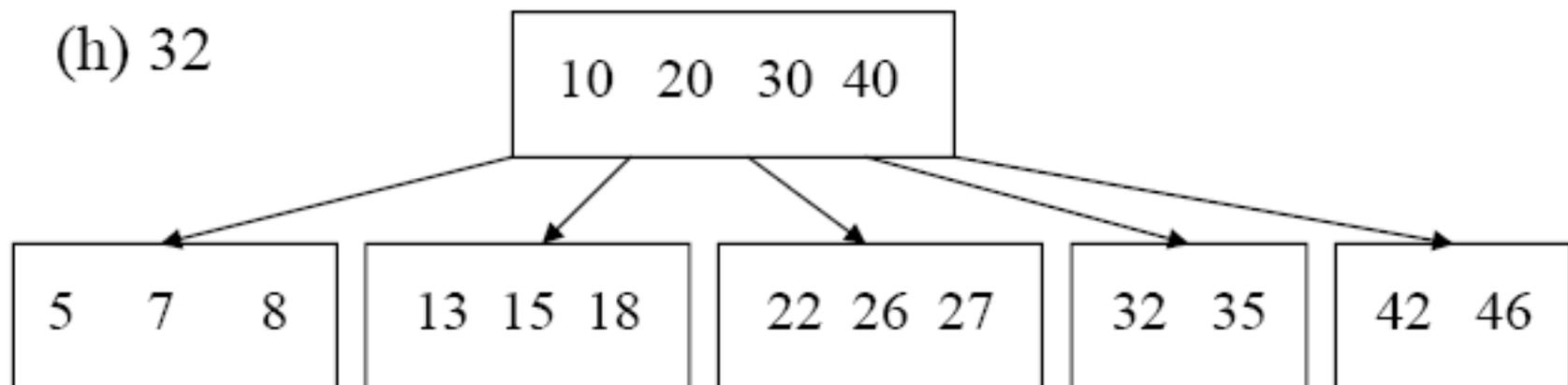
(f) 5



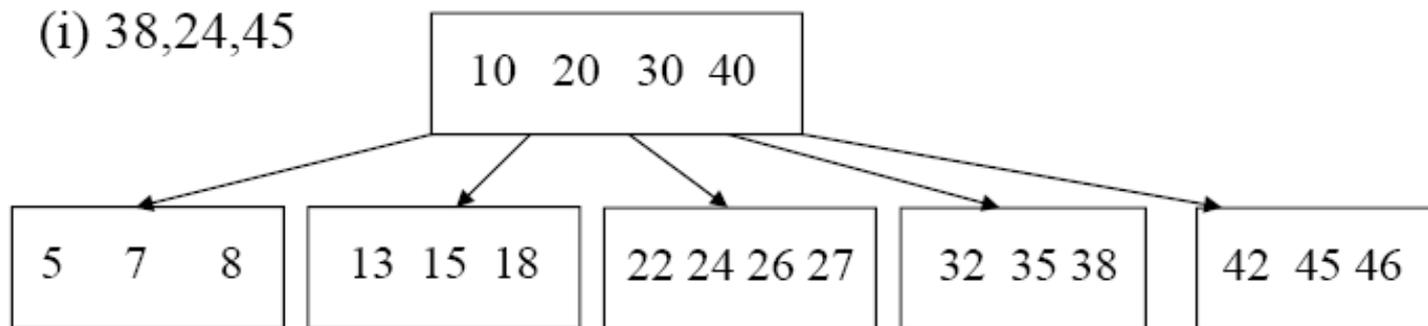
(g) 42,13,46,27,8



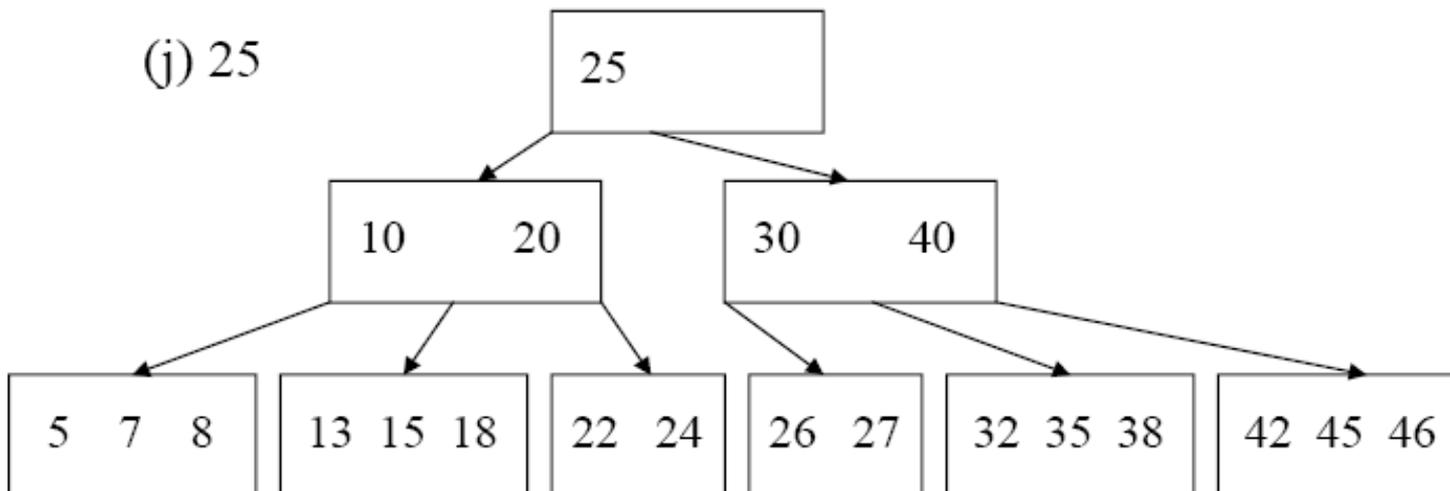
(h) 32



(i) 38,24,45



(j) 25



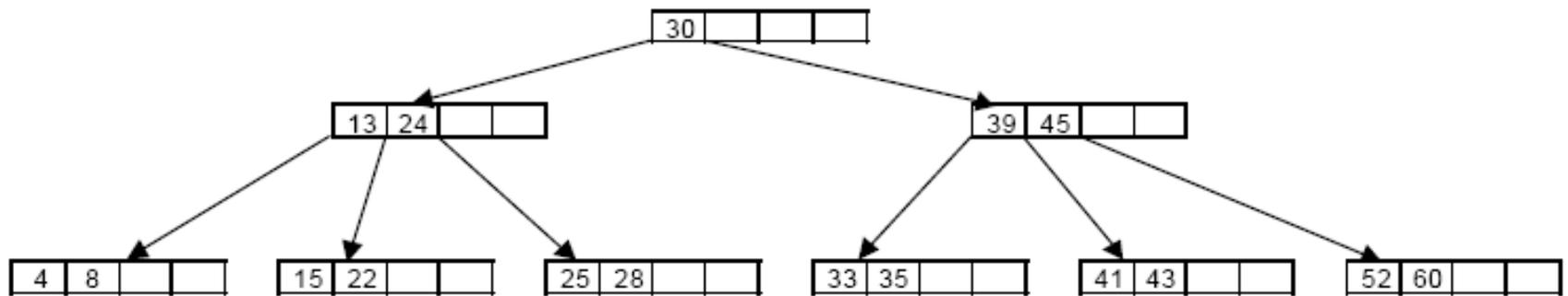
Eliminación en un árboles B

- Se pueden distinguir 2 casos:
- 1. El elemento a suprimir se encuentra en una página hoja:
 - Proceso de eliminación inmediato.
- 2. El elemento no se encuentra en una página hoja
 - Se usa la misma estrategia que en los árboles binarios de búsqueda:
 - Tomar el elemento mas a la derecha de la página hoja del subárbol izquierdo, o el de más a la izquierda de la página hoja del subárbol derecho.
- Para resolver el caso 2:
 - 2.1. Bajar por los apuntadores situados al extremo derecho hasta la página hoja P.
 - 2.2. Reemplazar el elemento a eliminar con el extremo derecho de P y reducir en 1 el tamaño de P.
 - 2.3. Verificar el n^o de elementos de la página:
 - No puede ser que $m < n$ por ser árbol B (**Subocupación**).

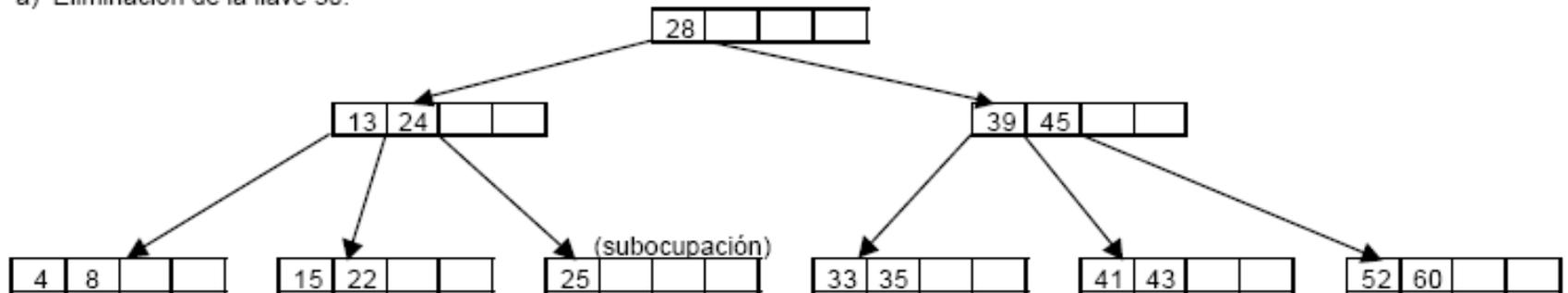
Eliminación

- En cualquier caso, es imprescindible que como mínimo cada página tenga n elementos.
 - Si tras la eliminación este número es inferior, se produce *subocupación*, que puede ser resuelta mediante la estrategia habitual de rebalanceo de páginas.
 - En este caso consiste en tomar los elementos de dos páginas contiguas (criterio: la de la izquierda) más el que las apunta desde la página madre, combinarlos en una sola página, y
 - Si existe *sobreocupación* se sube el elemento central.
-

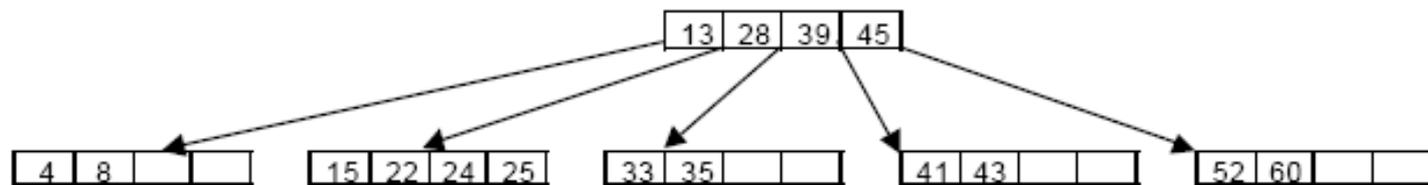
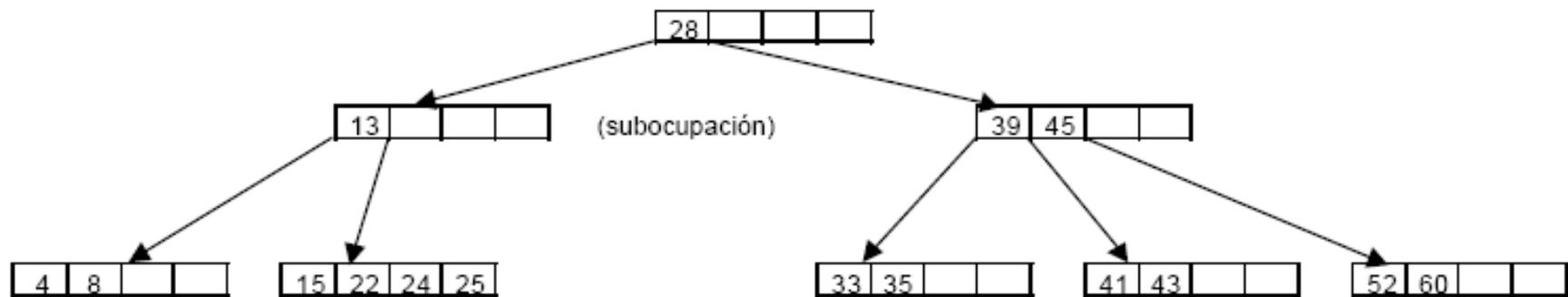
Ejemplo: vamos a eliminar consecutivamente las llaves 30, 35 y 60



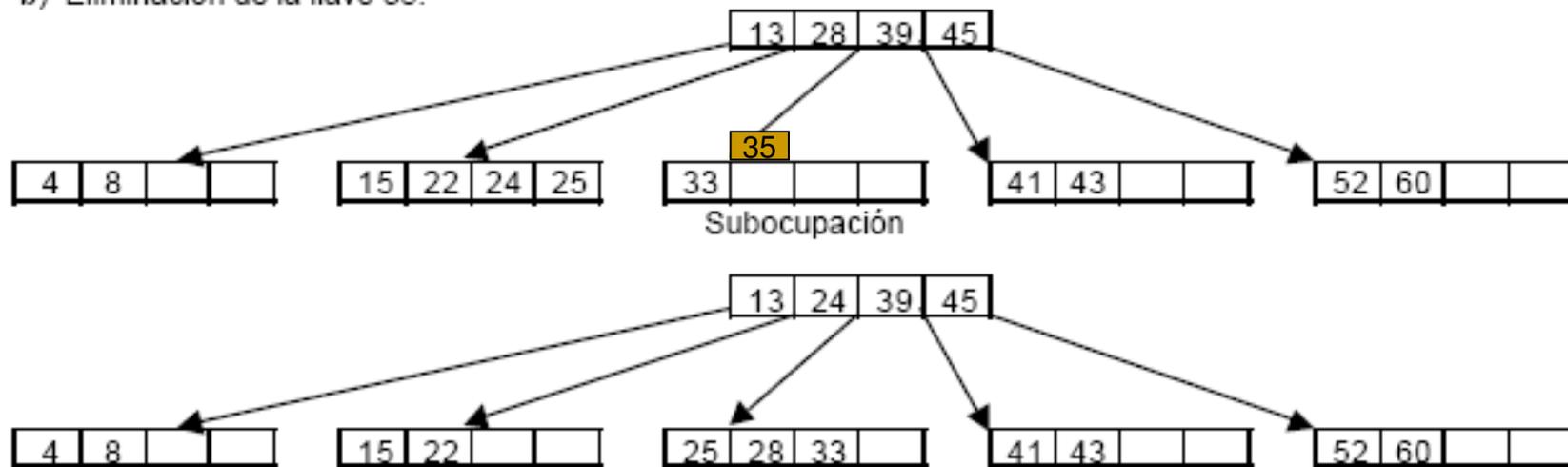
a) Eliminación de la llave 30.



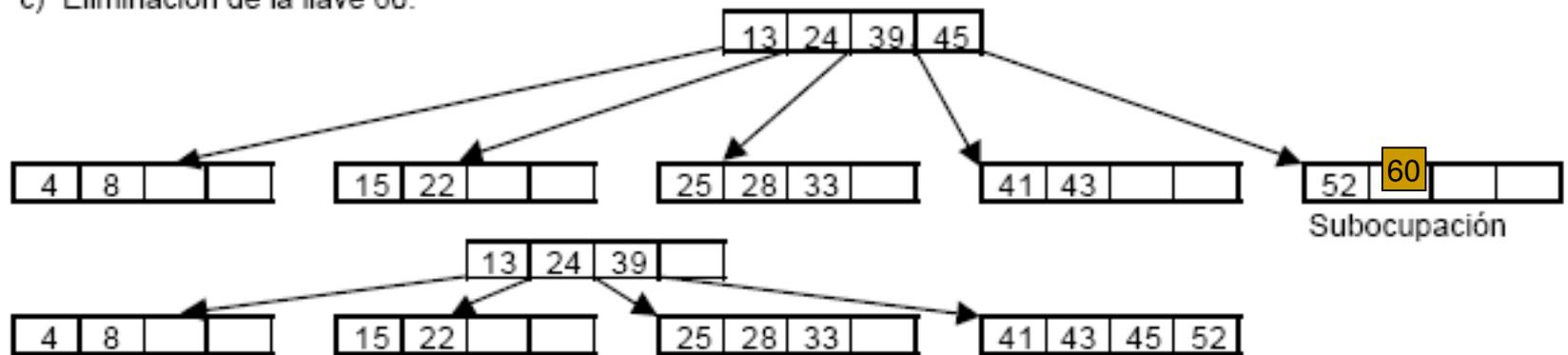
Se produce subocupación en la página que contenía la llave 28



b) Eliminación de la llave 35.

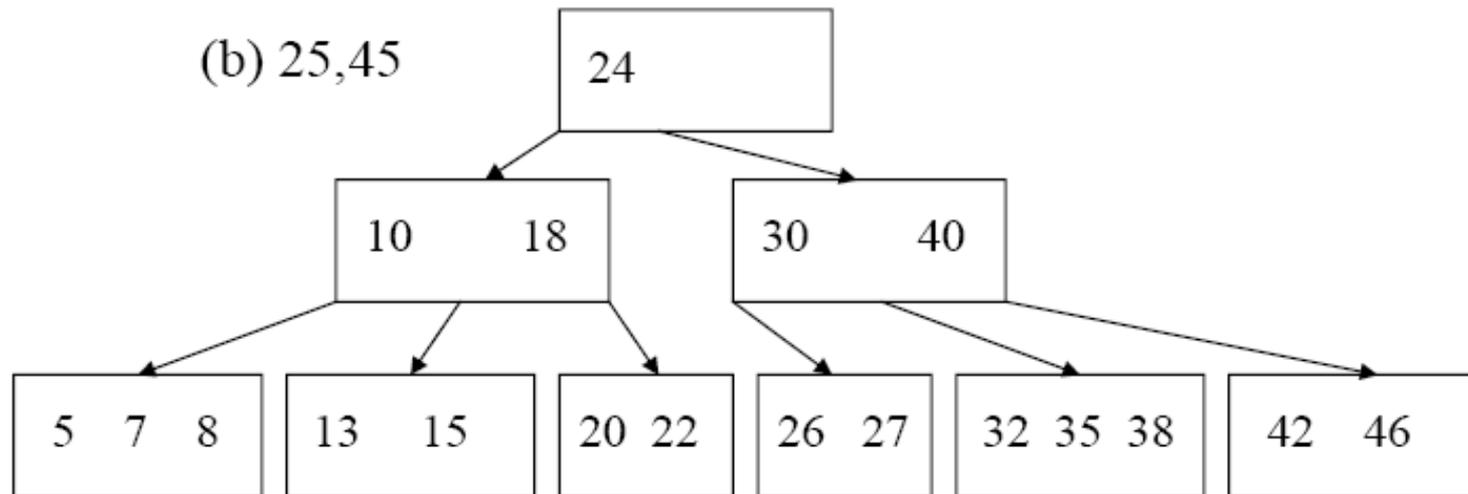
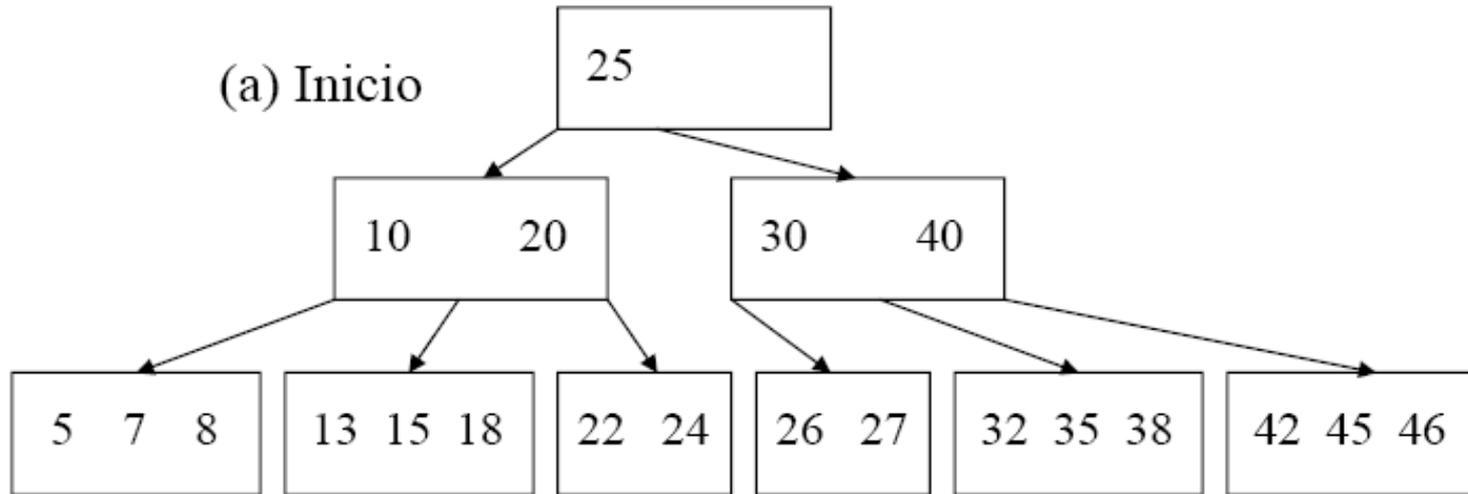


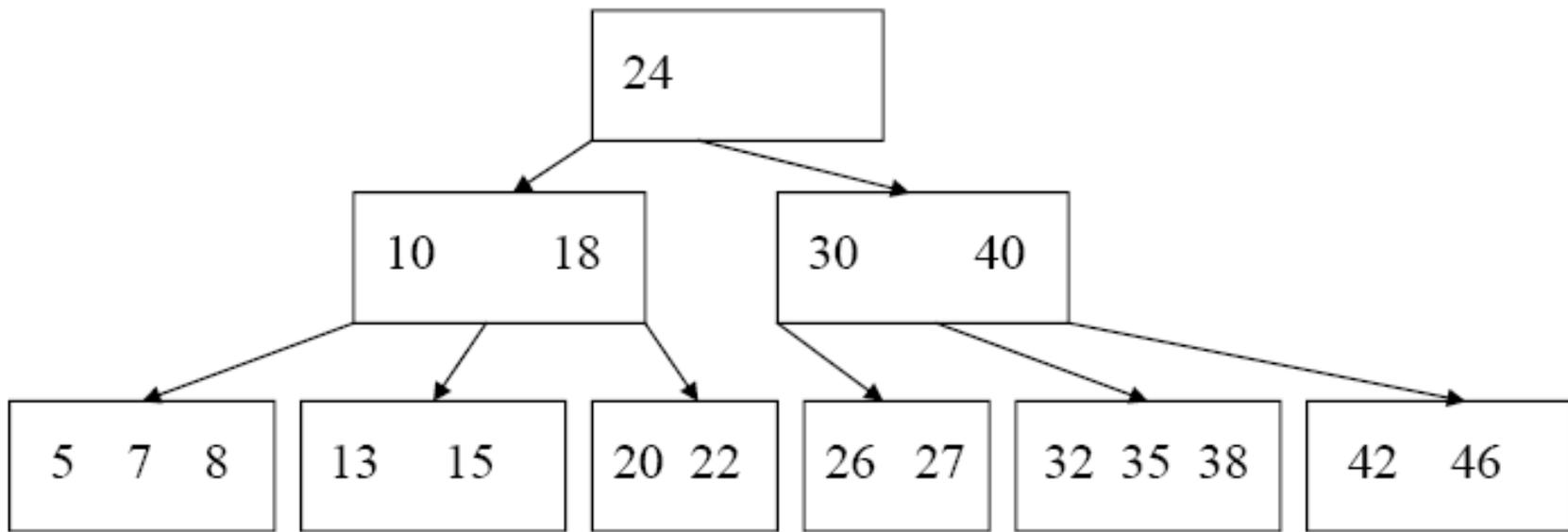
c) Eliminación de la llave 60.



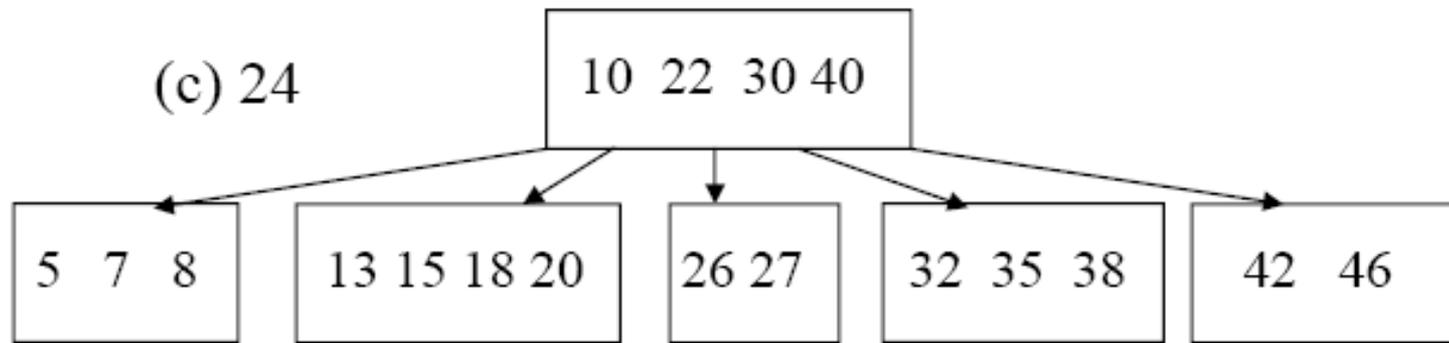
Ejemplo: Eliminación secuencial de las llaves:

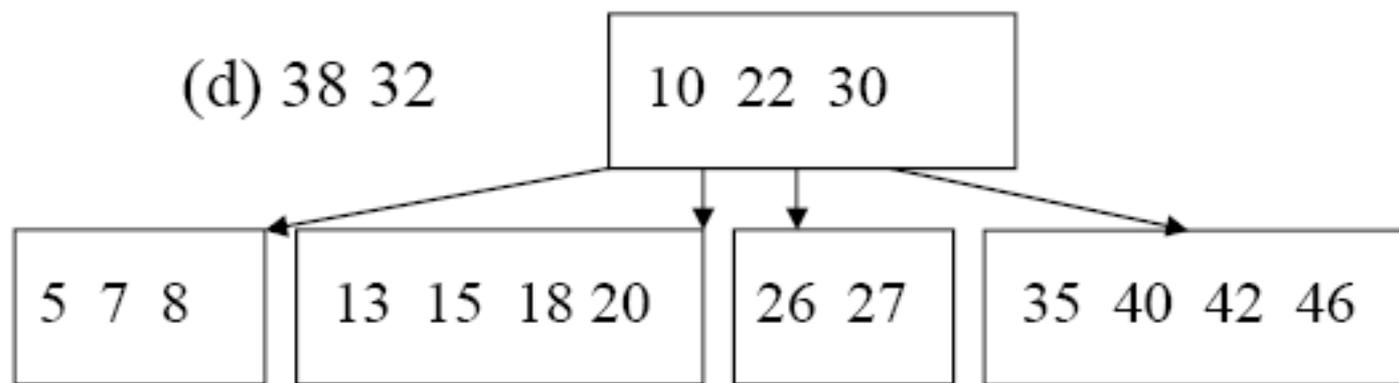
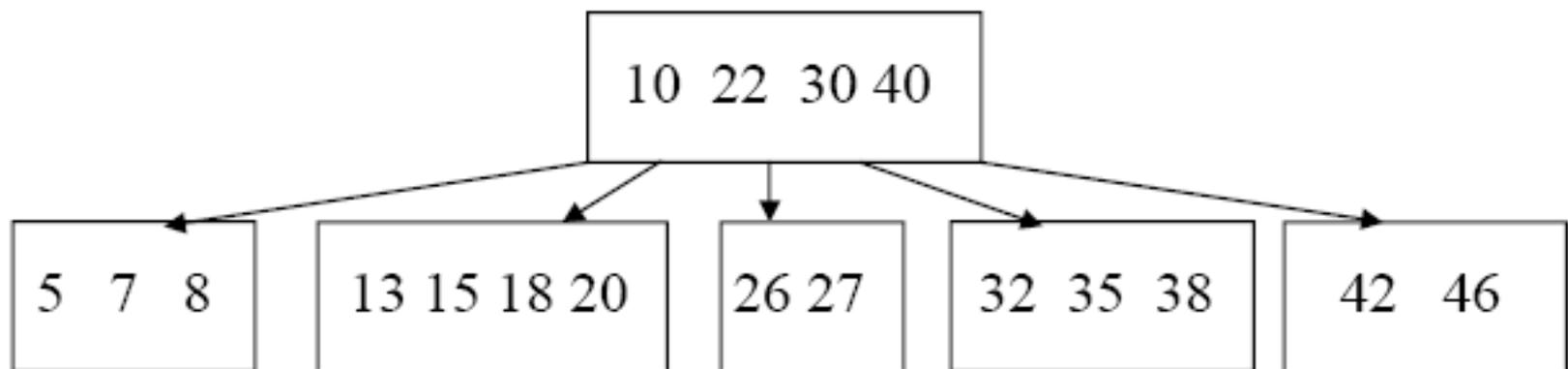
25, 45, 24, 38, 32, 8, 27, 46, 13, 42, 5, 22, 18, 26, 7, 35, 15

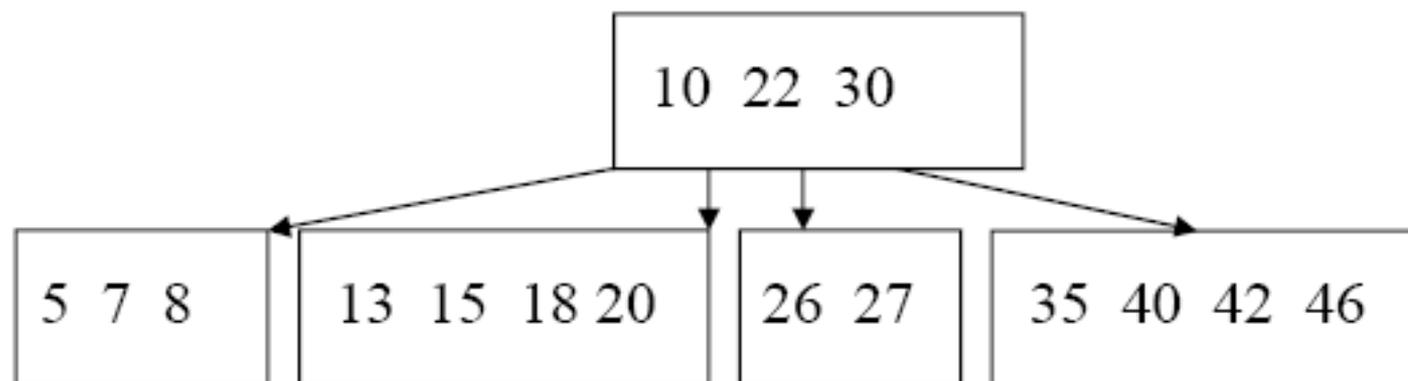




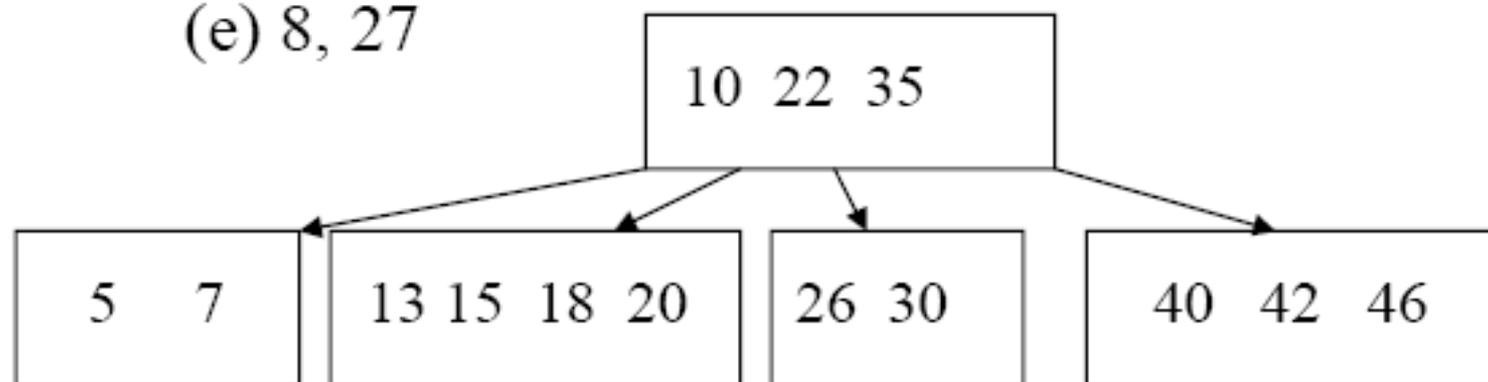
(c) 24



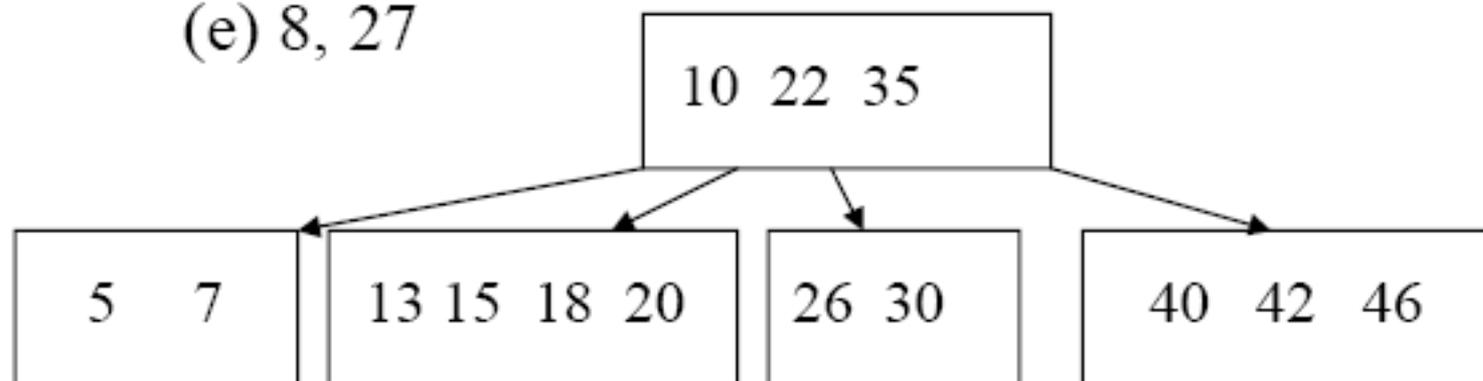




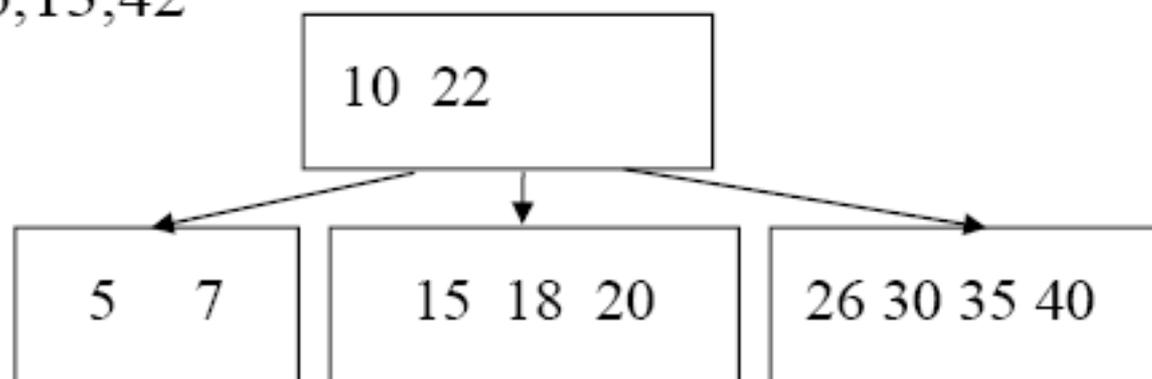
(e) 8, 27

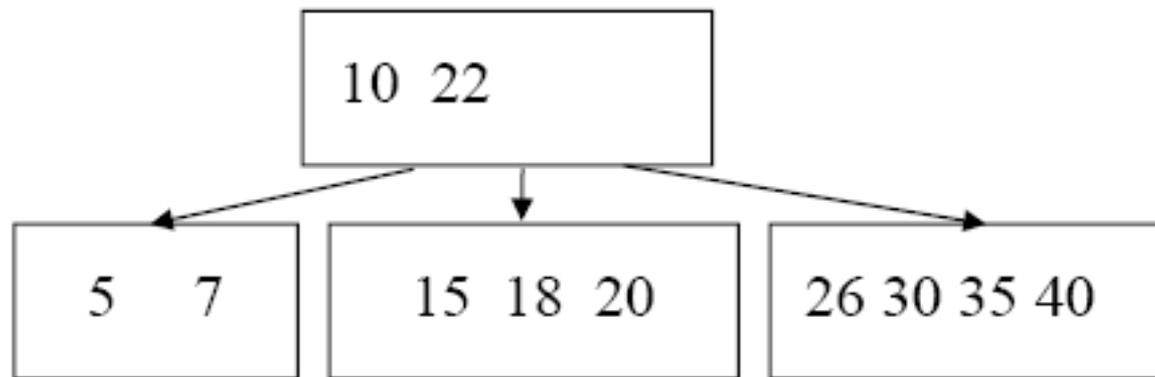


(e) 8, 27

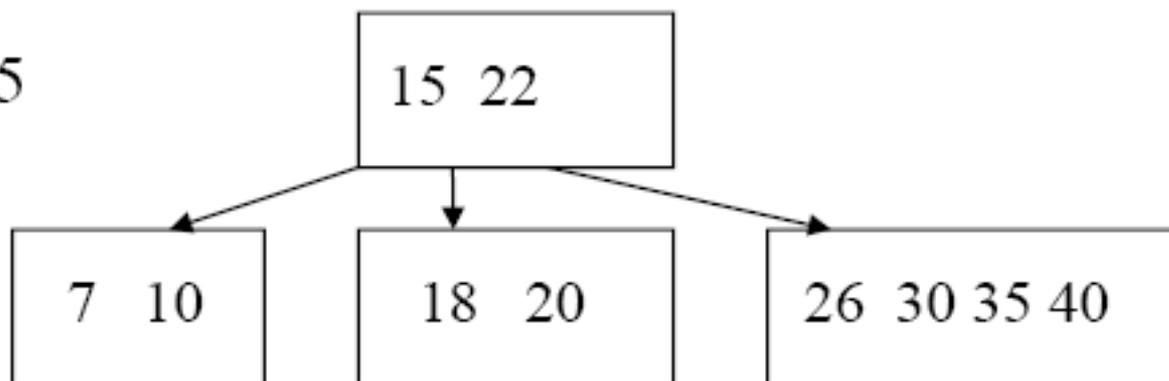


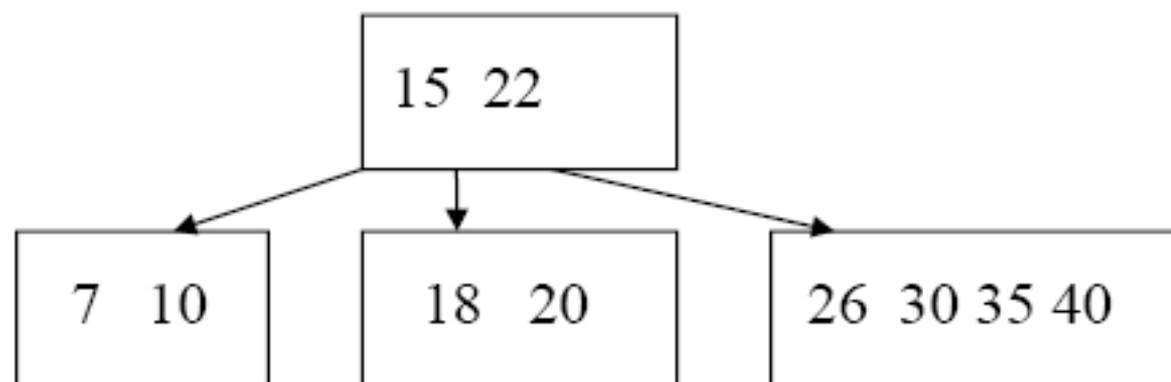
(d) 46,13,42



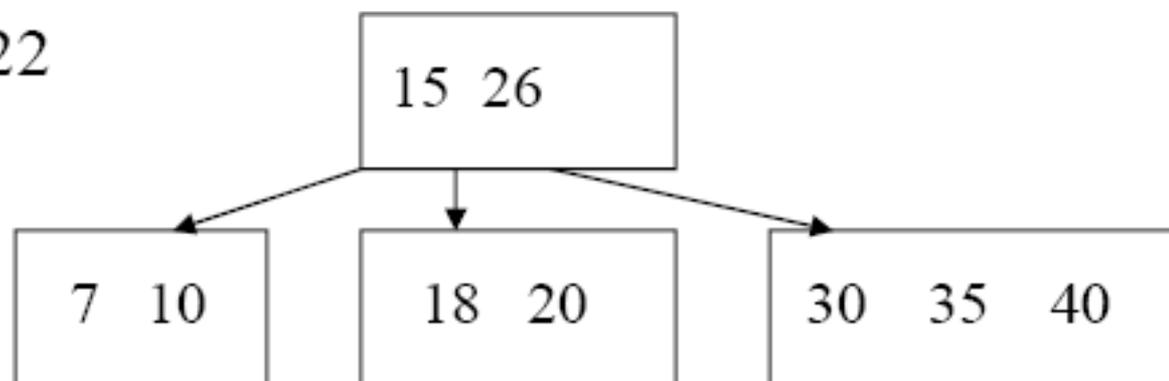


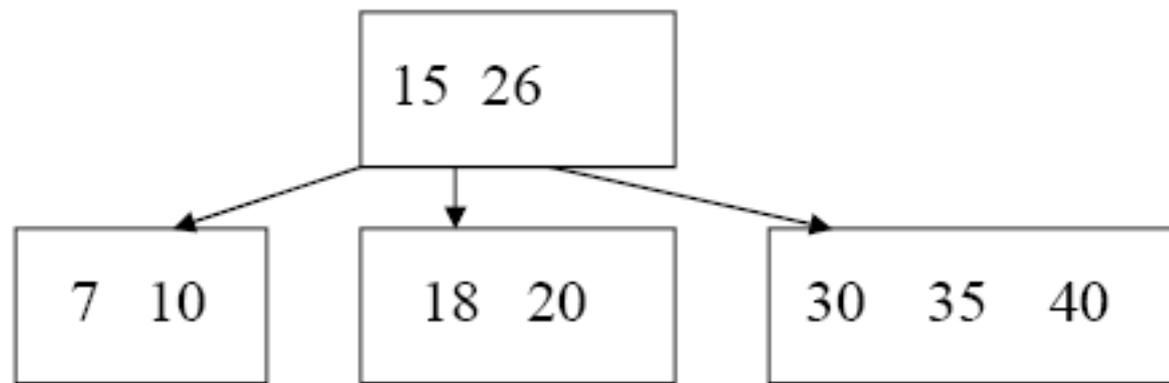
(e) 5



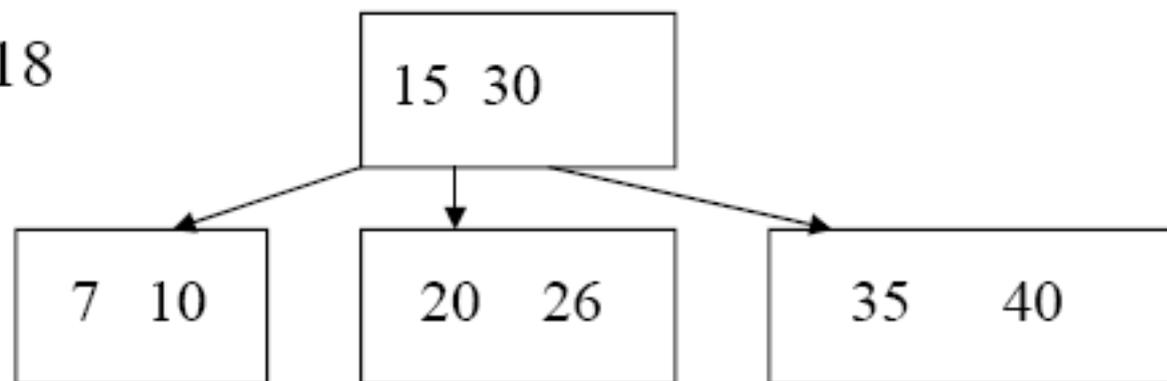


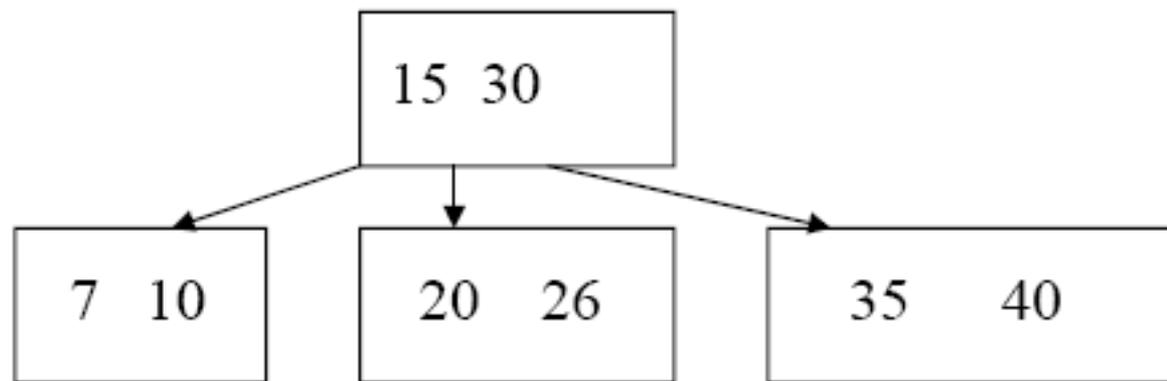
(f) 22



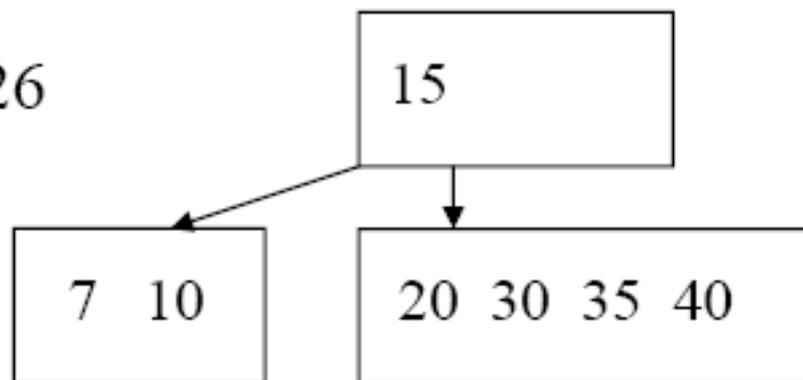


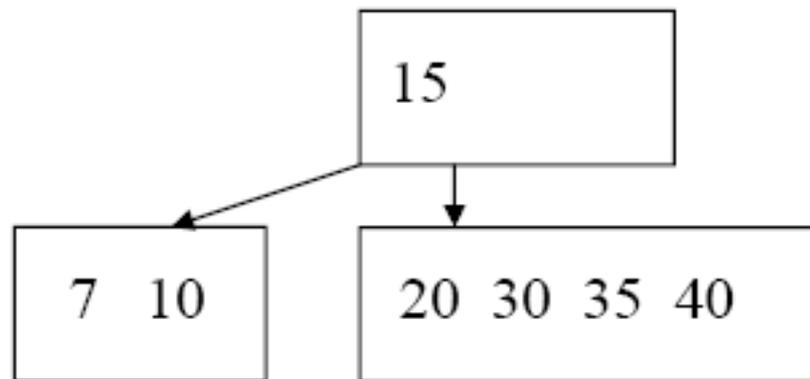
(g) 18



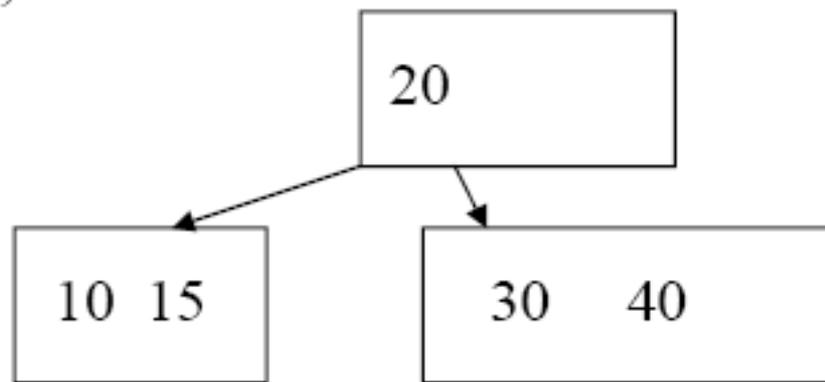


(h) 26

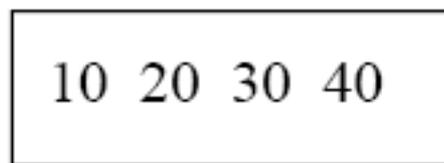




(e) 7, 35



(f) 15



6.3.- Árboles B binarios

- Un árbol B binario (árbol BB) es un árbol B de orden 1.
- Consta de nodos (páginas) con uno o dos elementos
 - Cada página contiene 2 o 3 punteros a los descendientes (árbol 2-3).
- Todas las páginas hoja aparecen en el mismo nivel (árbol B) y todas las páginas no hoja tienen 2 ó 3 descendientes (incluida la raíz).
- Se utilizan fundamentalmente en memoria primaria. En memoria secundaria tendrían una alta penalización de acceso.
- Hay dos tipos de apuntadores:
 - 1) A los descendiente
 - Verticales (izq. y dcho.)
 - 2) A los hermanos
 - Horizontales tiene un sentido único, hacia la derecha.

Representación de nodos de árbol BB

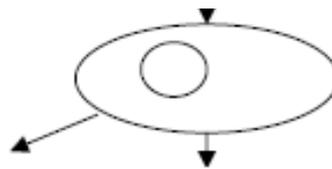
- Definición de un nodo de árbol

```
TYPE Ptr_Nodo = POINTER TO Nodo;  
TYPE Node = RECORD
```

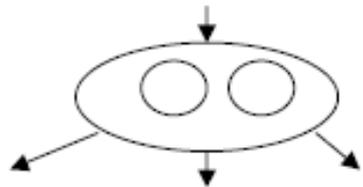
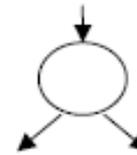
```
  llave: INTEGER;  
  izq, dch: Ptr_Nodo;  
  h: BOOLEAN (*rama horiz. a la derch.*)
```

```
END
```

$$L_{max} = 2 \lceil \log N \rceil$$

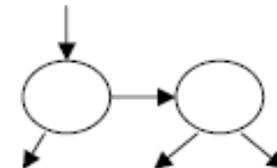


a



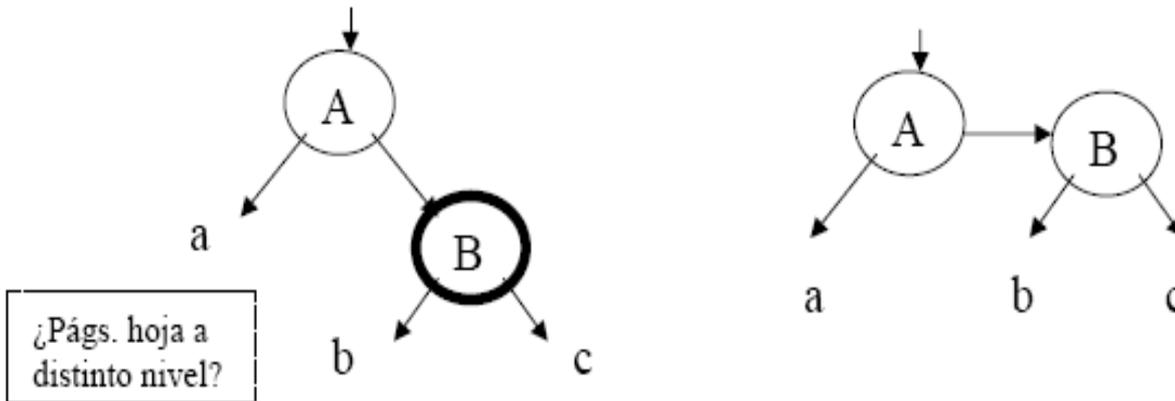
a

c



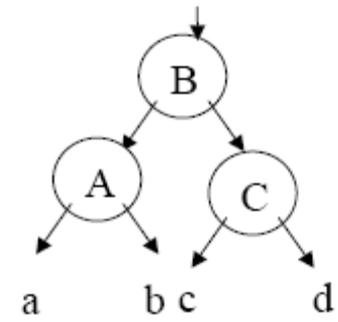
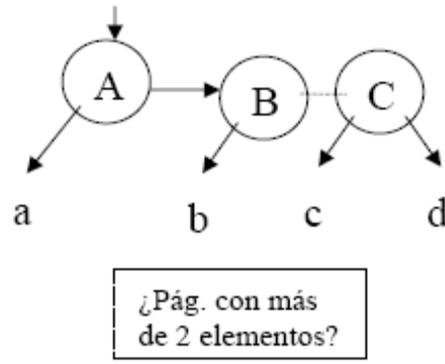
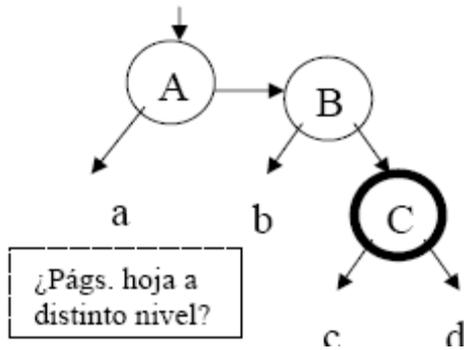
Inserción en árboles BB (I)

- 1) Crece el subárbol a la derecha de un nodo A, siendo A la única llave en su página



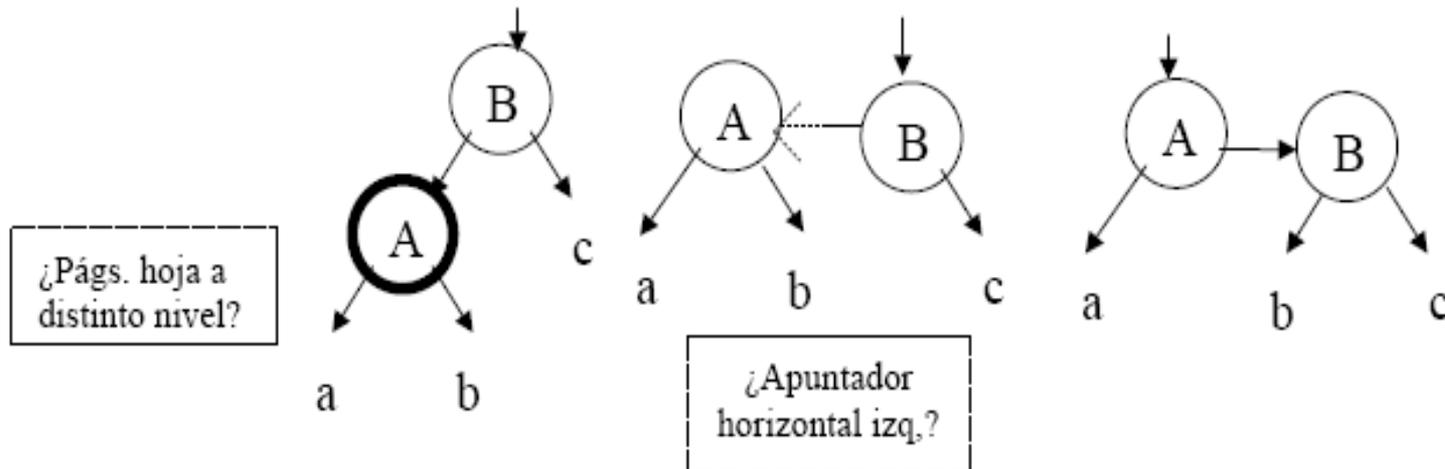
Inserción en árboles BB (II)

- 2) Crece el subárbol a la derecha de un nodo que posee dos hermanos A y B
 - Página de 3 nodos que hay que dividir.



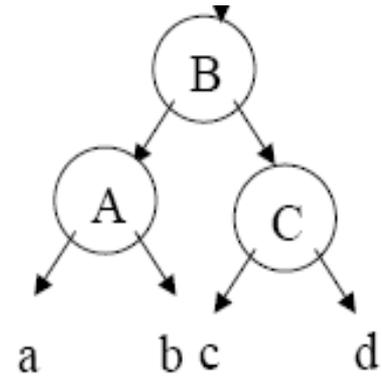
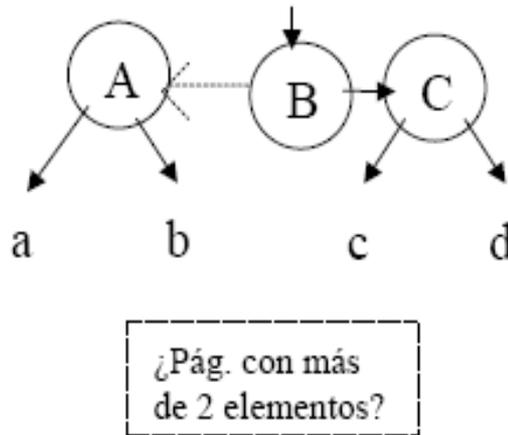
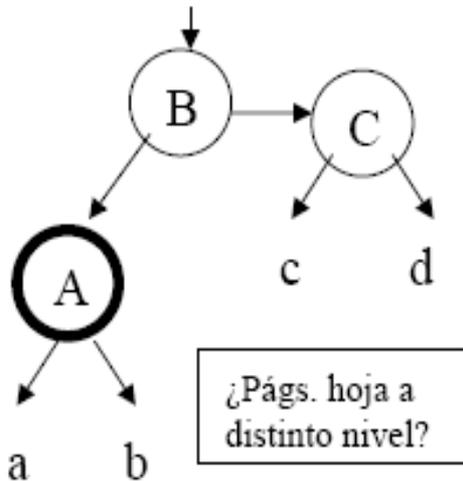
Inserción en árboles BB (III)

- 3) Crece el subárbol a la izquierda de un nodo B, siendo B la única llave en su página
 - Rotación de apuntadores



Inserción en árboles BB (IV)

- 4) Crece el subárbol a la izquierda de un nodo que tiene dos hermanos B y C
 - Página de 3 nodos que hay que dividir.



Arbol B binario Simétrico (árbol BBS)

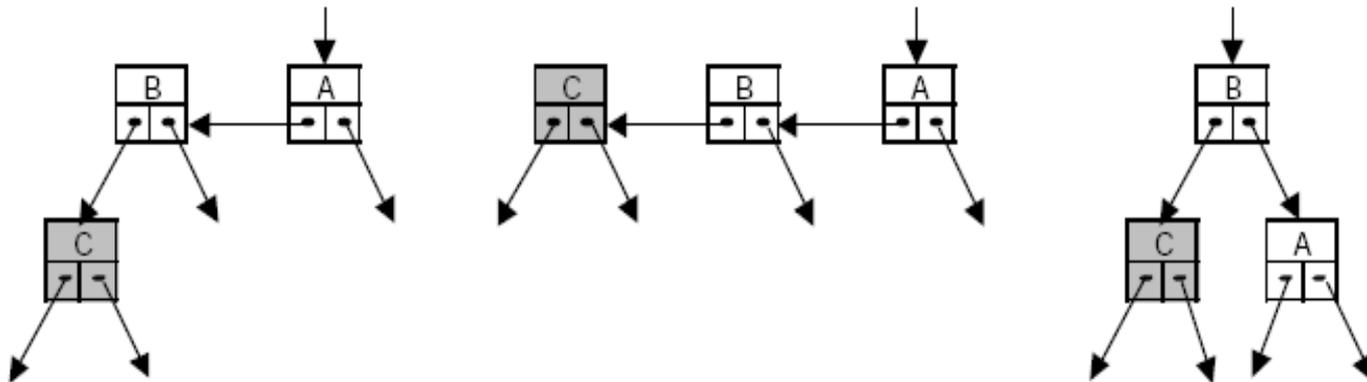
- Surgen con la idea de garantizar un crecimiento simétrico de los subárboles derecho e izquierdo del árbol.
 - Se definen como árboles de búsqueda que satisfacen las siguientes propiedades:
 - 1. Todo nodo contiene una llave, y como máximo dos apuntadores a subárboles.
 - 2. Todo apuntador es horizontal (derecho o izquierdo) o vertical. No existen dos apuntadores consecutivos horizontales (en el mismo sentido) en ninguna trayectoria de búsqueda. En cambio, si se permite que un nodo presente un apuntador derecho e izquierdo.
 - 3. Todos los nodos terminales aparecen al mismo nivel.
-

Arbol B binario Simétrico (árbol BBS)

- En general conducen a árboles de búsqueda ligeramente más eficientes pero los algoritmos de inserción y eliminación son más complejos.
- Un árbol BBS con n nodos no puede tener una altura superior a $\log n$, por tanto, la longitud de trayectoria de búsqueda es como máximo $L_{max} = 2\lceil \log n \rceil$

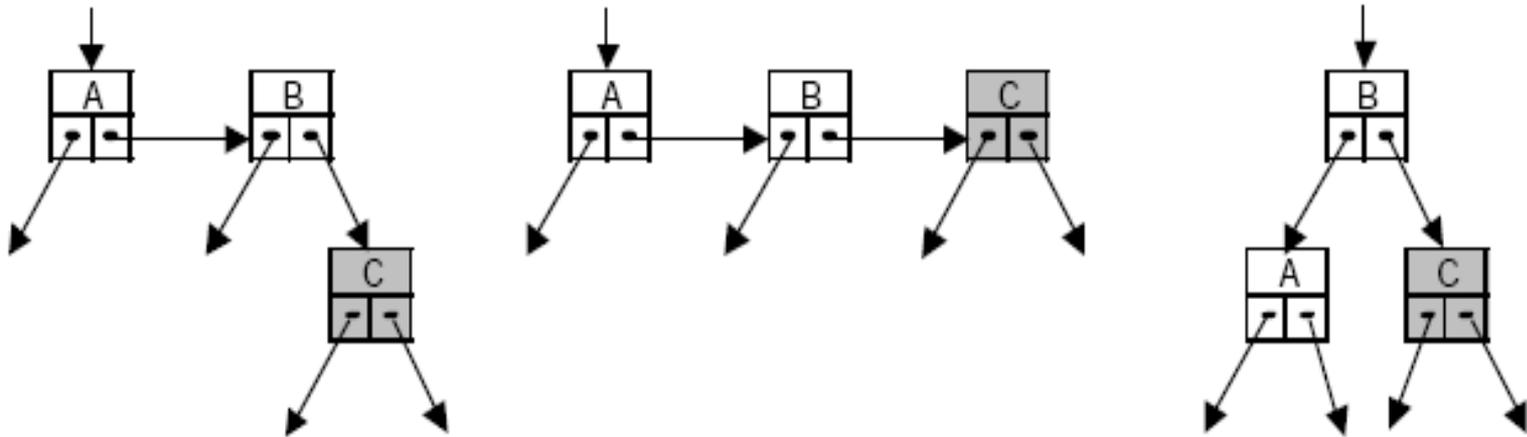
Inserción en árboles BBS

- En la inserción en árboles BBS puede darse cuatro casos en los que es necesario rebalanceo
 - 1.- Inserción y rebalanceo LL en un árbol BBS



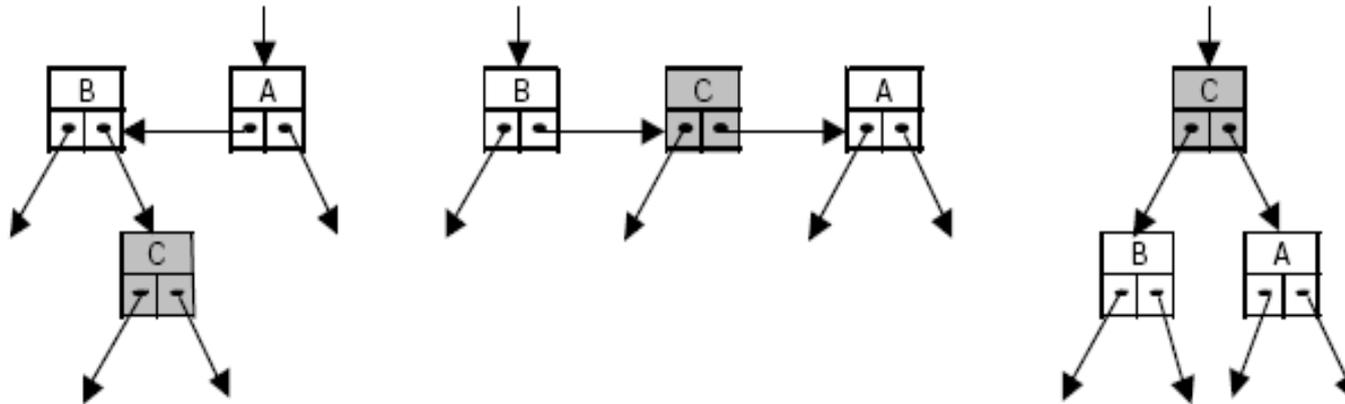
Inserción en árboles BBS

- 2) Inserción y rebalanceo RR en un árbol BBS



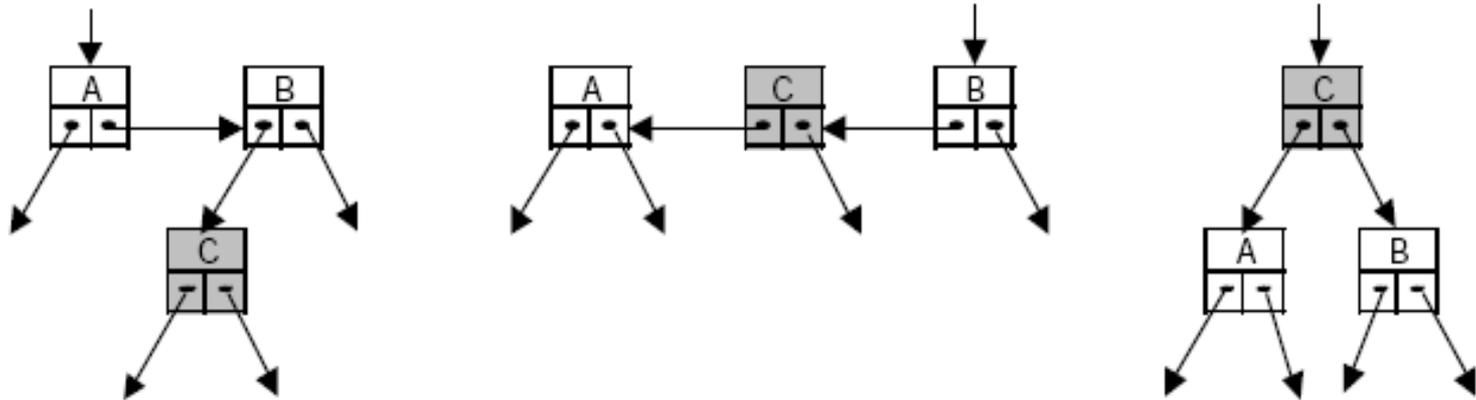
Inserción en árboles BBS

- 3) Inserción y rebalanceo LR en un árbol BBS



Inserción en árboles BBS

- 4) Inserción y rebalanceo RL en un árbol BBS



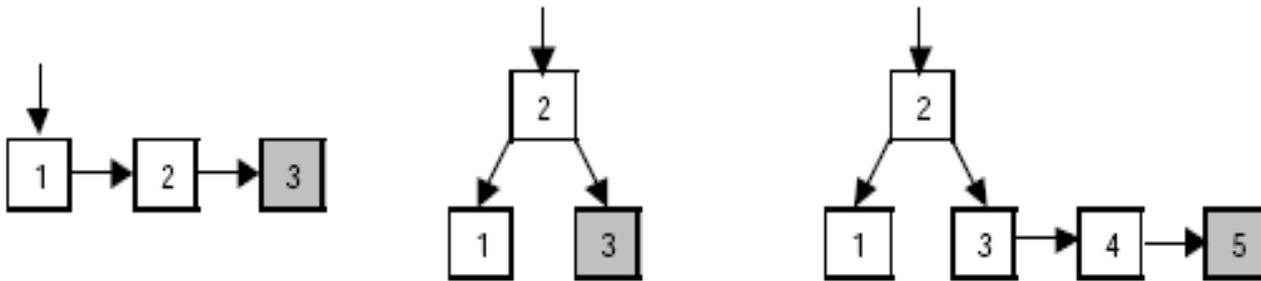
La declaración en Modula2 del nodo de un árbol BBS

```
TYPE Ptr_Nodo = POINTER TO Nodo;  
TYPE Nodo = RECORD  
    llave : INTEGER;  
    cont : CARDINAL  
    izq, dch : Ptr_Nodo;  
    h_izq, h_dch : BOOLEAN (*rama horizontal*)  
END
```

- la variable entera h indicar el crecimiento del árbol con el siguiente significado:
 - 1. $h = 0$:
 - el subárbol p no requiere cambios en la estructura
 - 2. $h = 1$:
 - el nodo p ha obtenido un hermano
 - 3. $h = 2$:
 - el subárbol p ha aumentado de altura

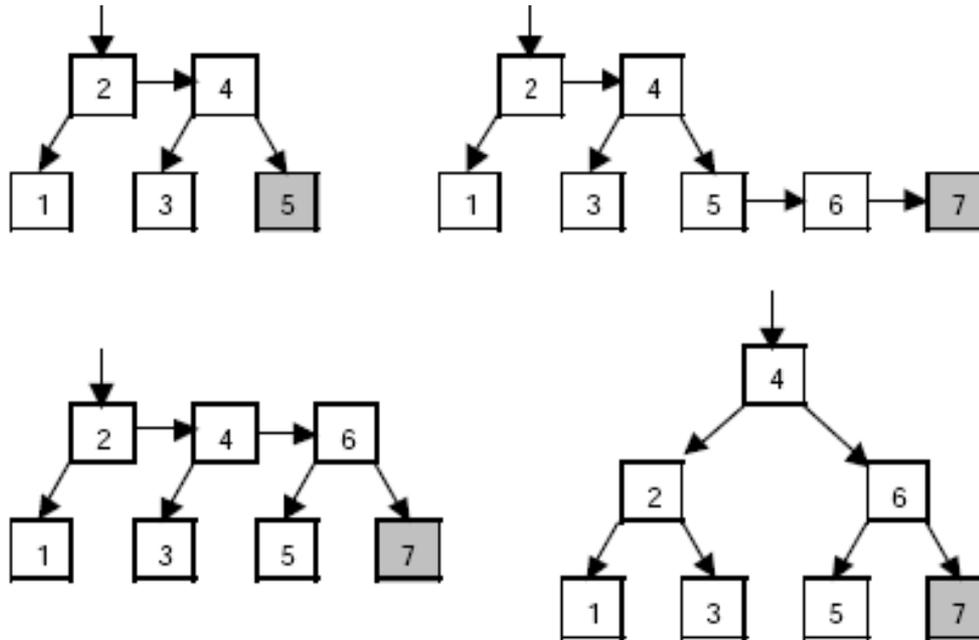
Ejemplos de crecimiento de árboles BBS

- Ejemplo1: Supongamos que en un árbol BBS inicialmente vacío se insertan consecutivamente las llaves: 1, 2, 3, 4, 5, 6, 7.



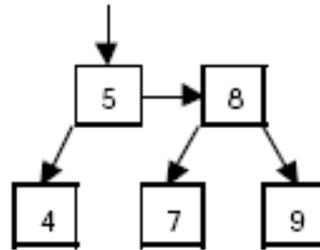
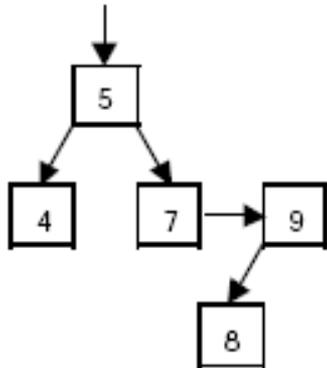
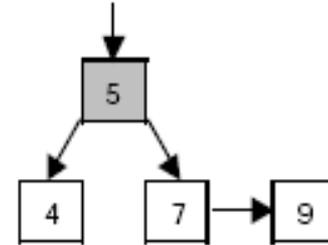
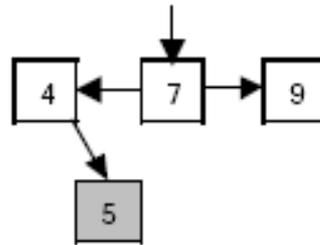
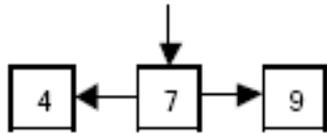
Ejemplos de crecimiento de árboles BBS

- insertan consecutivamente las llaves: 1, 2, 3, 4, 5, **6**, **7**



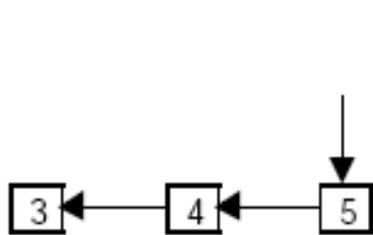
Ejemplos de crecimiento de árboles BBS

- Ejemplo2: Supongamos ahora que partiendo nuevamente de un árbol BBS inicialmente vacío se insertan consecutivamente las llaves: 7, 4, 9, 5, 8.

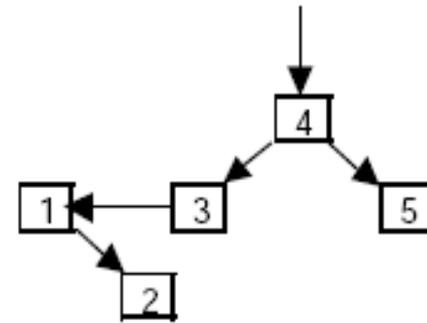
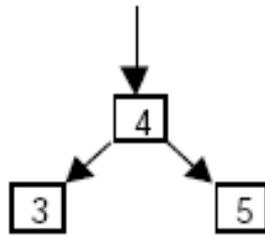


Ejemplos de crecimiento de árboles BBS

- Ejemplo3: Supongamos ahora que partiendo nuevamente de un árbol BBS inicialmente vacío se insertan consecutivamente las llaves: 5, 4, 3, 1, 2, 7, 6.



Se incumple condición 2
Subimos nodo central un nivel

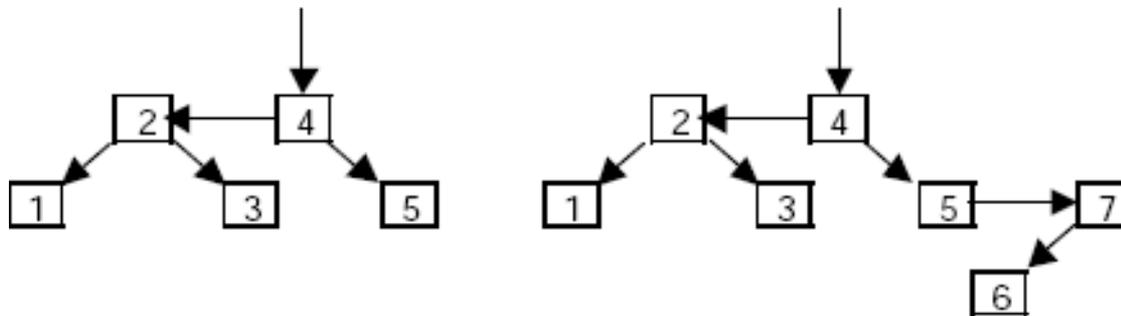


Se incumple condición 3
Subimos nodo 2 de nivel

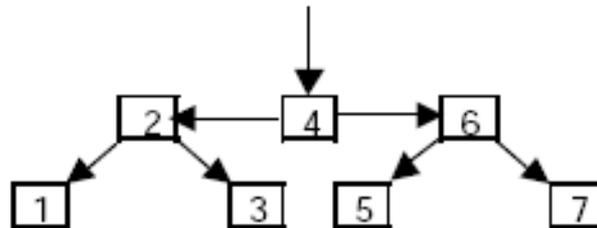
Ejemplos de crecimiento de árboles BBS

- se insertan consecutivamente las llaves: 5, 4, 3, 1, 2, 7, 6.

Al subir el nodo 2 se incumple condición 2 (nodos 1-2-3) por lo que se vuelve a subir de nivel el nodo central (2). Se cambia el enlace izquierdo del nodo raíz.



Se procede de manera análoga para el nodo 6 y se tiene



Ejemplos de crecimiento de árboles BBS

- Ejemplo4: Supongamos ahora que partiendo nuevamente de un árbol BBS inicialmente vacío se insertan consecutivamente las llaves: 4, 6, 2, 1, 7, 3, 5.

