

Arboles B Y MUCHO MAS



Tipazo

Insercion Arboles B

In order

¿L/E necesarias para el alta de la clave 80?

L2, L1, E1, E4, E2 (cada nodo se lee a lo sumo 1 vez)

Propiedades de un Árbol B de orden M

- Cada nodo del árbol puede contener como **máximo M descendientes** y **M-1 elementos**.
- La raíz no posee descendientes directos o tiene al menos dos.
- Un nodo con **X descendientes** directos contiene **X-1 elementos**.
- Todos los nodos (salvo la raíz) tienen como **mínimo $\lceil M/2 \rceil - 1$ elementos** y como **máximo M-1 elementos**.
- Todos los nodos terminales se encuentran al mismo nivel.

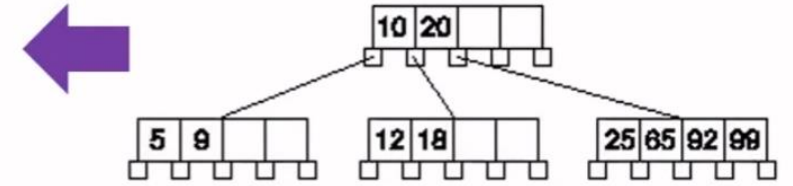
• Inserción

1. Buscar el nodo correspondiente en orden a la clave a insertar (siempre nodo hoja).
2. Si el nodo tiene espacio suficiente se inserta el valor siguiendo el orden de las claves.
3. Si el nodo donde se debe insertar la clave se encuentra lleno, ocurre "Overflow"
 - 3.1 El nodo lleno se divide en 2 repartiendo las claves entre ellos.
 - 3.2 La clave intermedia se coloca en el nodo padre (sube). Si dicho nodo no existe, se crea.

orden: 5

max: 4

min: 2



Orden de un arbol(M): cantidad de referencias q puede sacar, o cantidad de valores +1, del nodo

Max: (M-1).. 5-1 = 4 ese arbol como max tiene 4 elementos

Minimo: ($\lceil m/2 \rceil - 1$)... $\lceil 5-1 \rceil / 2$... $\lceil 4 \rceil / 2$... 2 (nos quedamos con la parte entera)

Caso impar

5, 7, 10, 15, 18

Caso par

70, 75, 80, 90

Izq: se queda en el nodo con Overflow

mid: promociona(sube)

Derecha: nuevo nodo

Izq: se queda en el nodo con Overflow

mid: seria derecha (el min) {80} promociona(sube)

Derecha: [90] nuevo nodo

En español...

Un árbol-B de orden 4 tiene 4 referencias, máximo 3 valores y mínimo 1
Un árbol-B de orden 5 tiene 5 referencias, máximo 4 valores y mínimo 2

Insercion Arboles B, Overflow ejemplo

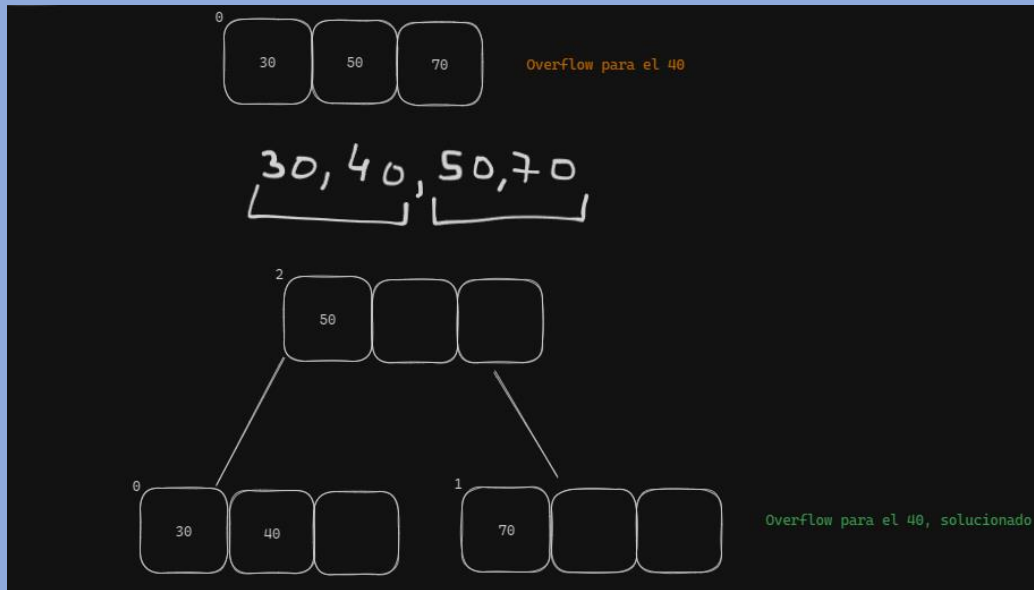
• Ejemplos

1.- Insertar en un árbol b de orden 4 las siguientes claves:

70,50,30,40,20,80,25,90,75,10,15

Orden 4
4 referencias
3 elementos max
1 elemento como minimo
In order

1

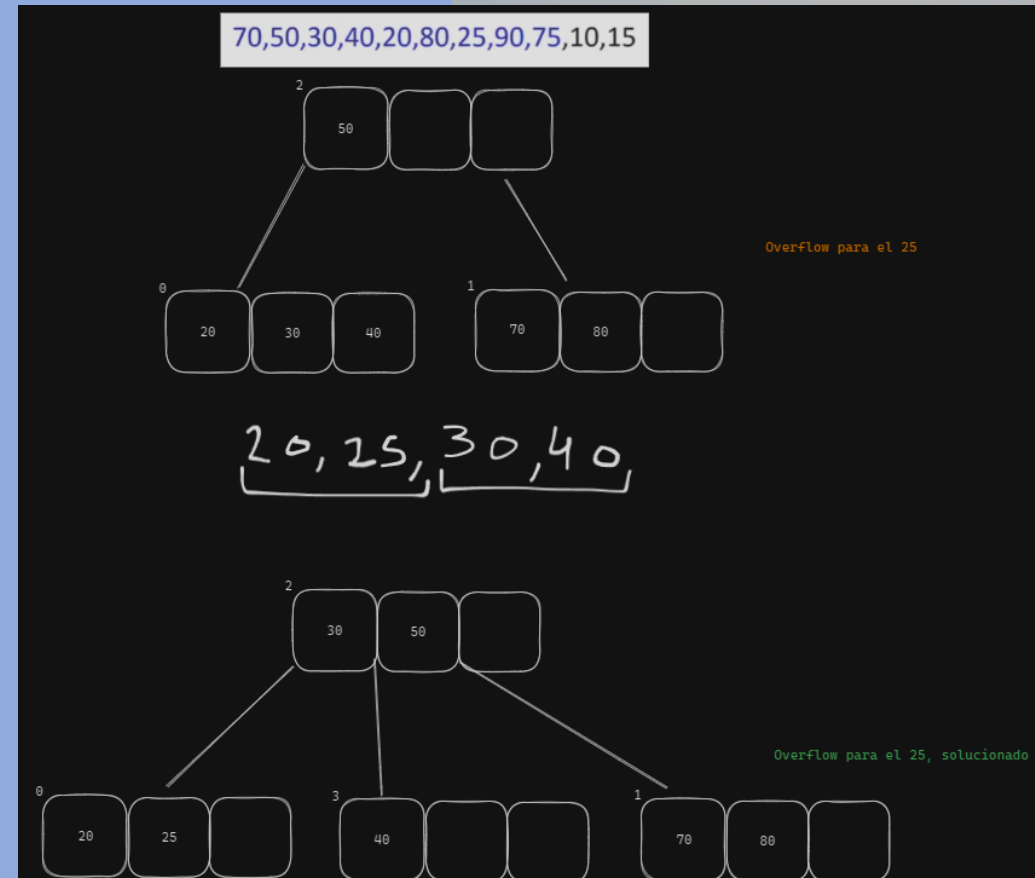


Overflow

Pasos a Seguir

- Se crea un nuevo nodo.
- La primera mitad de las claves se mantiene en el nodo con overflow.
- La segunda mitad de las claves se traslada al nuevo nodo.
- La menor de las claves de la segunda mitad se promociona al nodo padre.

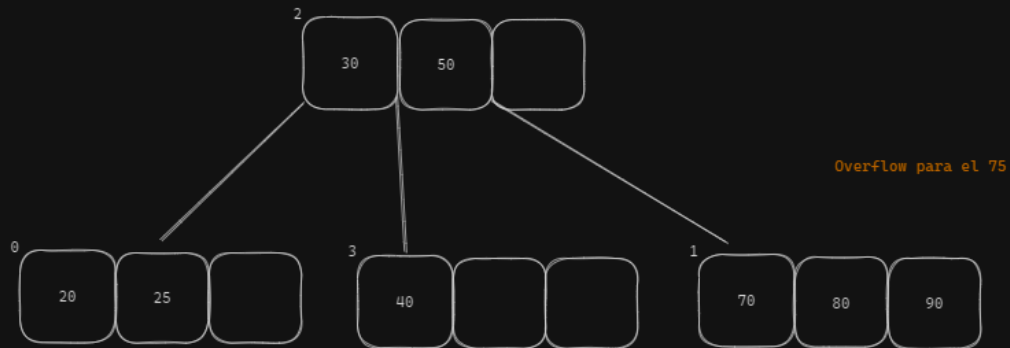
2



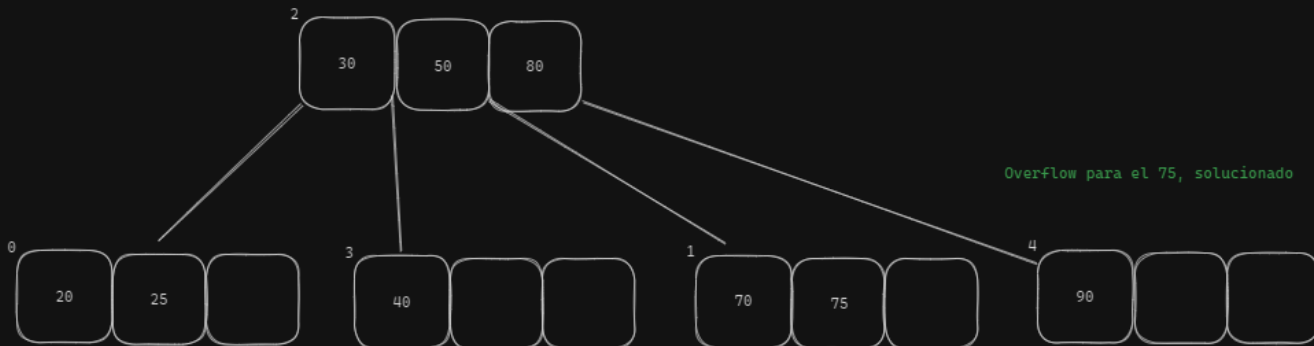
Siguiendo ejemplo

3

70,50,30,40,20,80,25,90,75,10,15

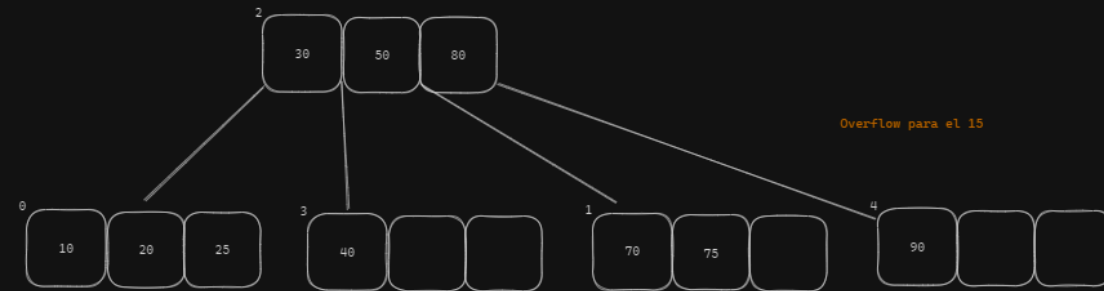


70, 75, 80, 90

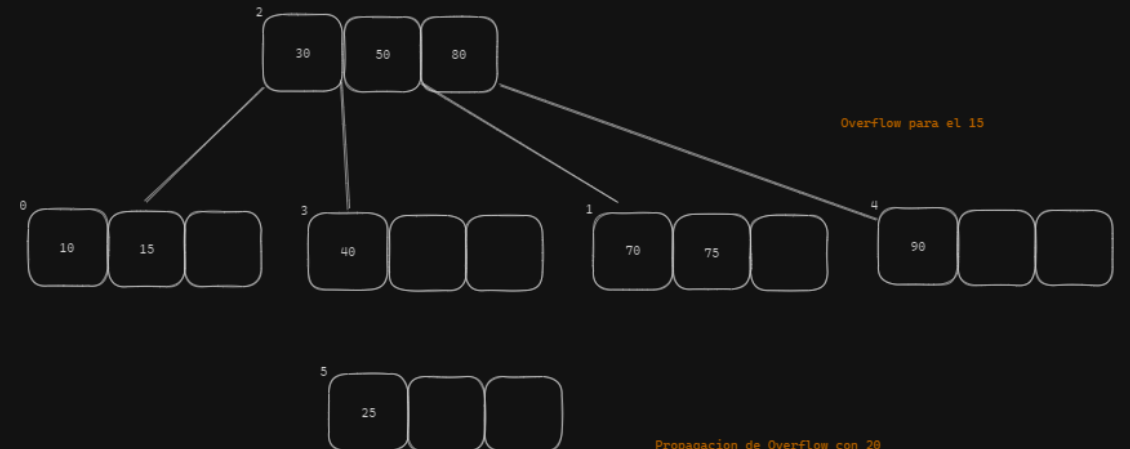


4

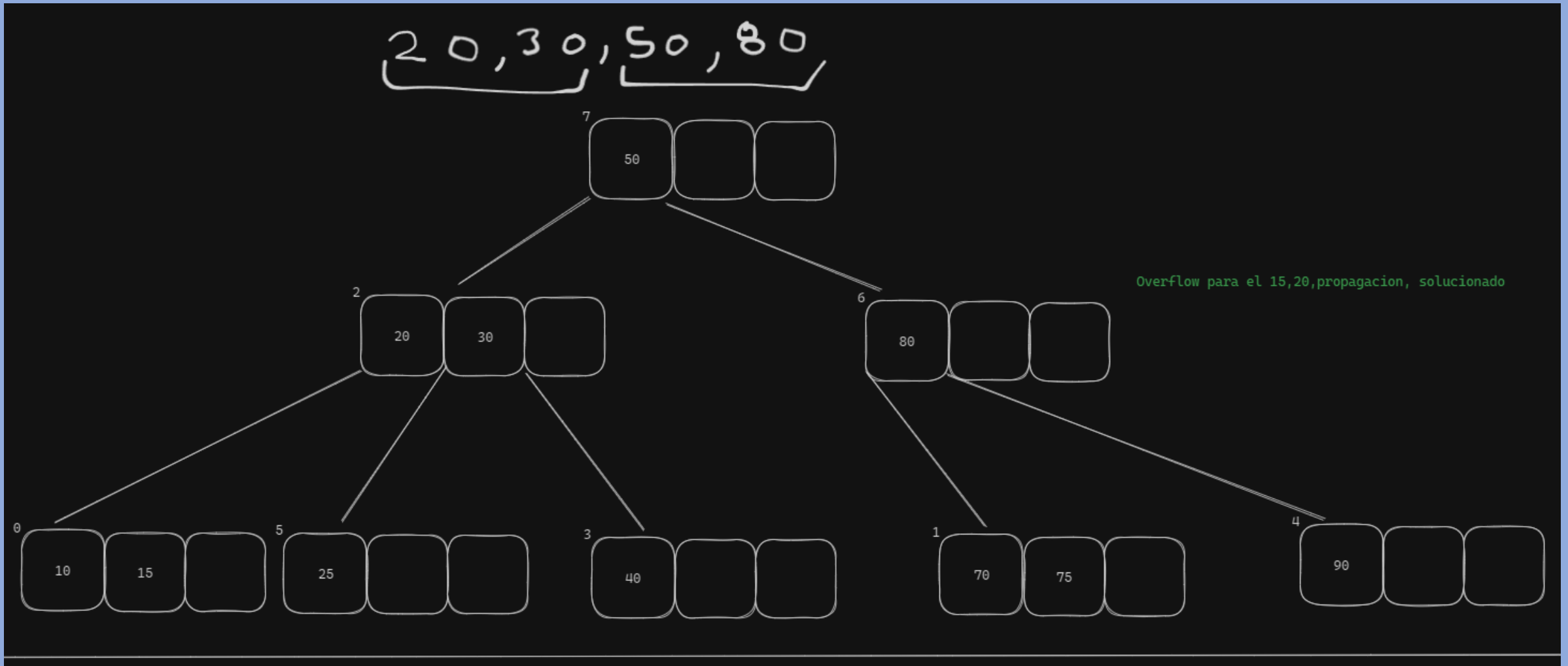
70,50,30,40,20,80,25,90,75,10,15



10, 15, 20, 25



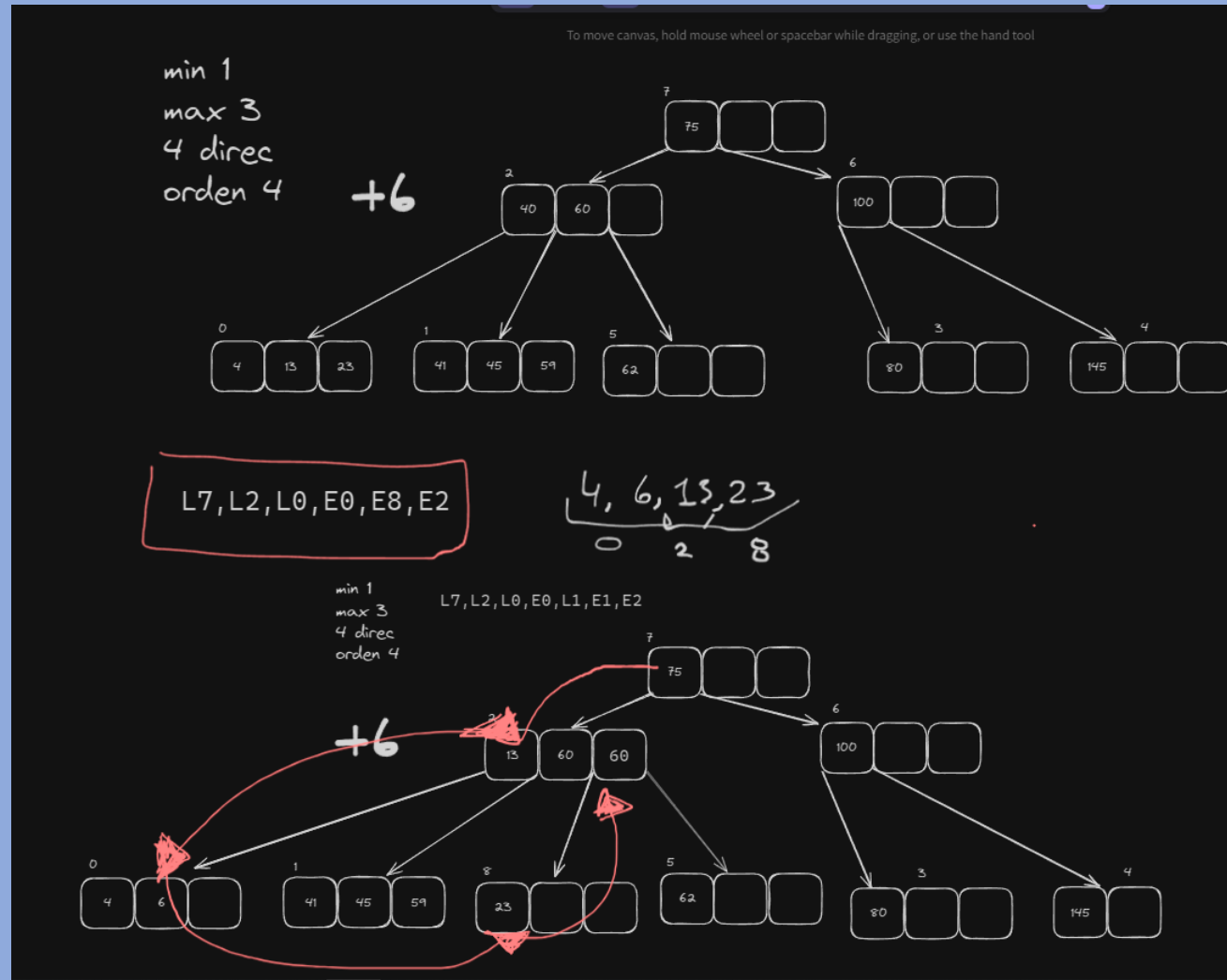
Siguiendo con el ejemplo



ESCRITURAS Y LECTURAS EN INSERCCION B

las L/E, tengo entendido que a un chabon le dijo un ayudante que se hacian:
De izquierda a derecha y de arriba hacia abajo (esto no aplica para las busquedas)

Si estoy en el proceso de
Overflow, creando el nodo
Derecho, las E/L es:
Leo izq, E y E derecho q va
Ser el nuevo nodo y después
arriba

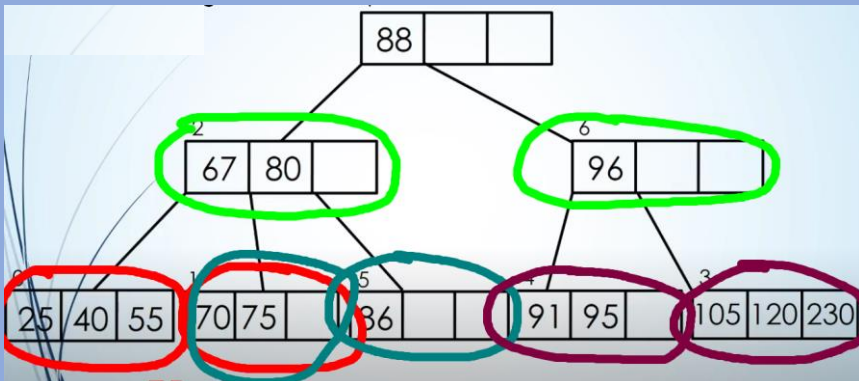


Bajas en Arboles B

Bajas

1. Si la clave a eliminar no está en una hoja, se debe reemplazar con la menor clave del subárbol derecho. **RAÍZ**
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. En caso contrario, se debe tratar el underflow

Hermanos adyacentes/inmediatos



Bajas - Underflow

4. Primero se intenta redistribuir con un hermano adyacente. La redistribución es un proceso mediante el cual se trata de dejar cada nodo lo más equitativamente cargado posible.
5. Si la redistribución no es posible, entonces se debe **fusionar** con el hermano adyacente.

Me tienen que dar la política con la q voy a tratar

Políticas para la resolución de underflow:

1. **Política izquierda:** se intenta redistribuir con el hermano adyacente izquierdo, si no es posible, se fusiona con hermano adyacente izquierdo.
2. **Política derecha:** se intenta redistribuir con el hermano adyacente derecho, si no es posible, se fusiona con hermano adyacente derecho.
3. **Política izquierda o derecha:** se intenta redistribuir con el hermano adyacente izquierdo, si no es posible, se intenta con el hermano adyacente derecho, si tampoco es posible, se fusiona con hermano adyacente izquierdo.
4. **Política derecha o izquierda:** se intenta redistribuir con el hermano adyacente derecho, si no es posible, se intenta con el hermano adyacente izquierdo, si tampoco es posible, se fusiona con hermano adyacente derecho.

Casos especiales UnderFlow

Políticas para la resolución de underflow:

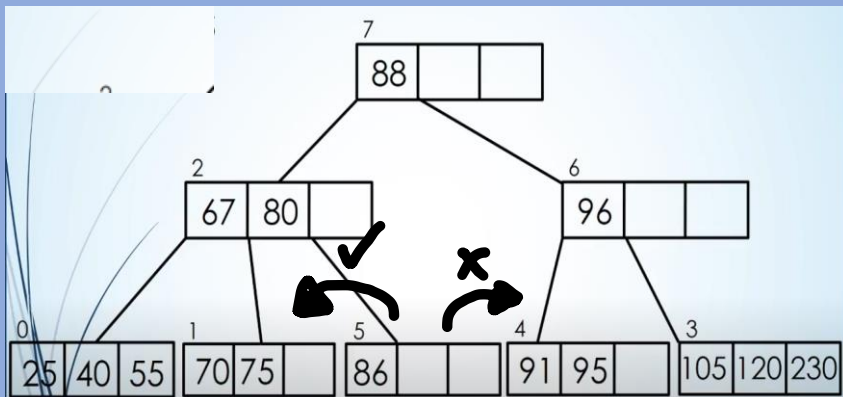
Casos especiales: en cualquier política si se tratase de un nodo hoja de un extremo del árbol debe intentarse redistribuir con el hermano adyacente que el mismo posea.

Aclaración:

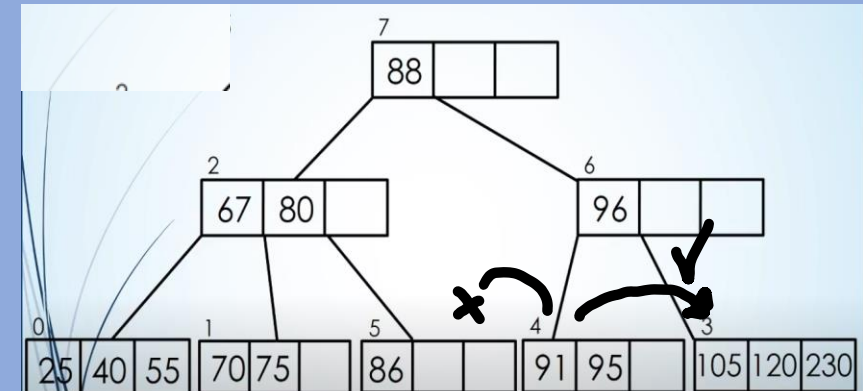
- En caso de underflow lo primero que se intenta **SIEMPRE** es redistribuir y el hermano adyacente se encuentra en condiciones de hacer la redistribución y no se produce underflow en el.

Casos hipoteticos de ejemplo

Tengo Underflow en el nodo 5, con politica de derecha.. Como no tiene tomo el nodo 1



Tengo Underflow en el nodo 4, con politica de izquierda.. Como no tiene hermano izquierdo, tomo el nodo 3



Ejemplardo Bajas, Arboles B

- **Eliminación de claves**

Regla de eliminación:** Si el orden del árbol es ***m, cada hoja debe tener al menos $(m/2) - 1$ claves.

1.- Si la clave a eliminar se encuentra en una hoja, se elimina directamente.

2.- Si al realizar la eliminación, el nodo mantiene el mínimo número de claves, finaliza, en caso contrario se debe hacer una ***redistribución***.

- **Eliminación de claves**

3.- Si el elemento no se encuentra en una hoja, se debe “subir” la clave que se encuentra más a la derecha en el sub árbol izquierdo, (o más a la izquierda del subárbol derecho)

4.- Si al subir esta clave, en la hoja respectiva no se cumple el mínimo número de claves, se debe realizar una ***redistribución***.

Redistribución

a) Si una hoja vecina inmediata tiene suficientes claves disponibles (préstamo) :

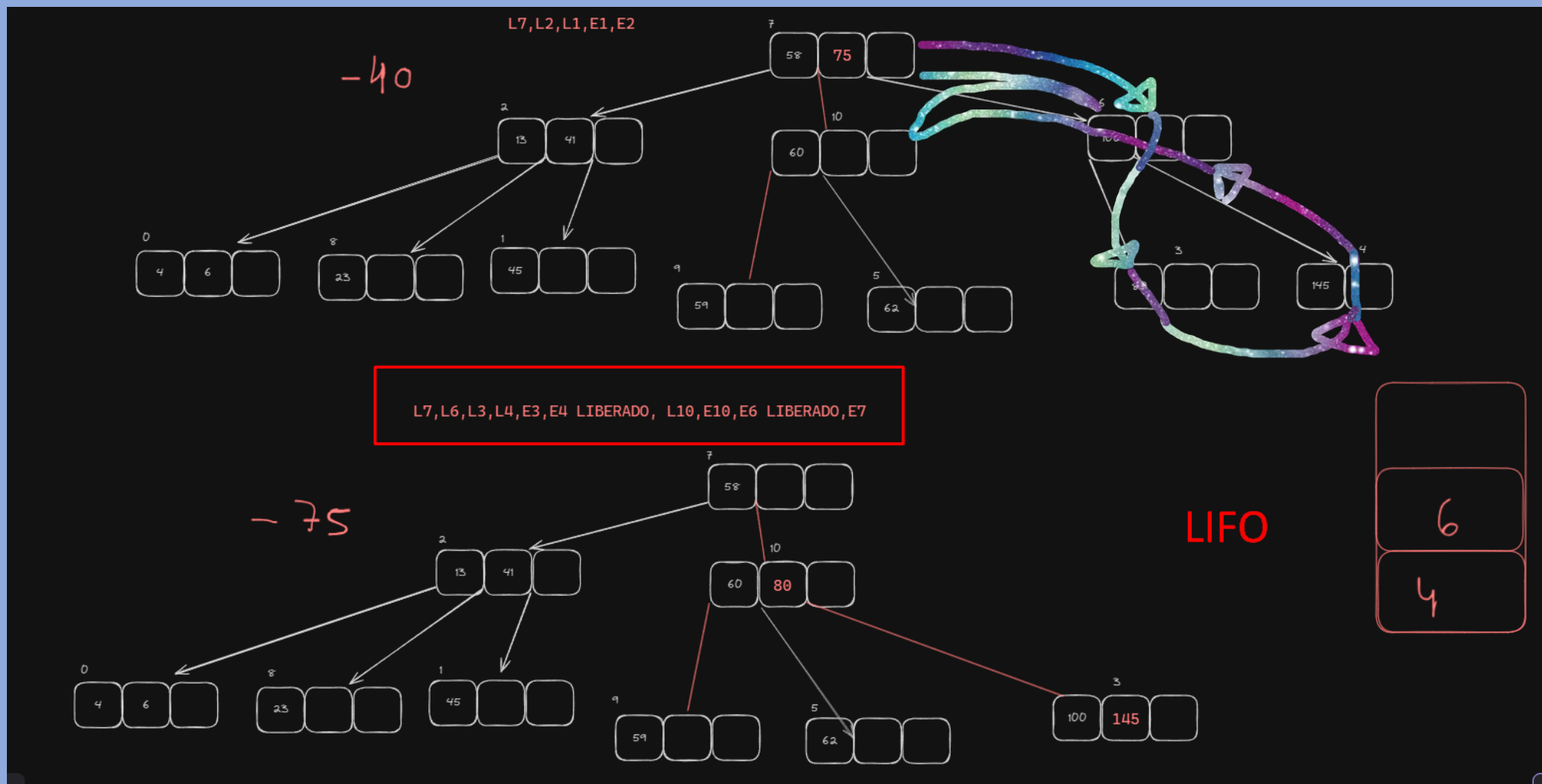
La clave que se encuentre más a la izquierda “sube”

La clave del nodo padre “baja”

b) Si no cuenta con suficientes claves, la hoja donde se ha eliminado la clave, la hoja adyacente y la clave padre se juntan en un solo nodo. (fusión)

Fusion: ver el hermano adyacente y fusionar el contenido del hermano con el separador del padre, el contenido del fusion, se queda en la izquierda por convencion

Escrituras lecturas en Bajas B



Va de izquierda a decha y despues arriba, menos en las busquedas.
Si o Si tiene que conocer(Lectura) con los nodos q va interactuar

Arbol B+

- ✓ Constituyen una mejora sobre los árboles B, pues conservan 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 facilitar el recorrido secuencial rápido. Cuando se lee en orden lógico, lista todos los registros por el orden de la clave.

Raiz = ?? Inicial = ??				
	punteros	clave (datos)	#reg	Sig
Nodo 0				

Solo tiene un puntero al siguiente, yo como nodo no conozco al anterior

Inserccion

Inserción B+

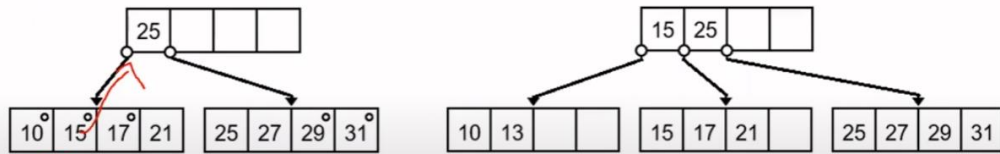
Dificultad: Inserción en nodo lleno (overflow).

El nodo afectado se divide en 2, distribuyéndose las claves **lo más equitativamente posible**. Una **copia** de la clave del medio o de la menor de las claves mayores (casos de overflow con cantidad pares de elementos) se promociona al nodo padre. El nodo con overflow se divide a la mitad.

► La copia de la clave sólo se realiza en un overflow ocurrido a nivel de hoja.

Caso contrario -> igual tratamiento que en árboles B.

Ejemplo1. Inserción de la clave 13



10, 13, 15, 17, 21

Cuando hay Overflow en las hojas, se copia el valor hacia arriba Y se deja el valor original en el hijo derecho.

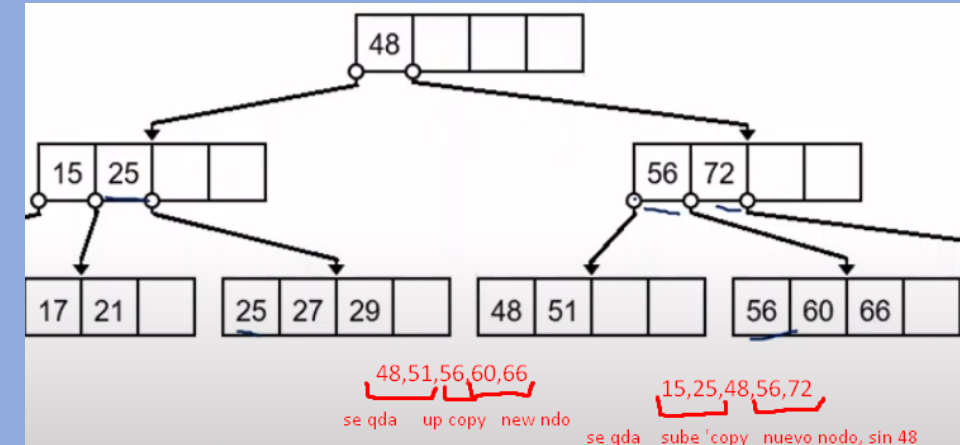
“Siempre la copia que sube se queda del lado derecho”



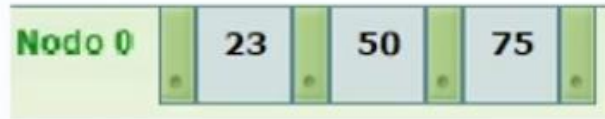
Cuando hay Overflow en un nodo intermedio, no se sube una copia, son referencias!

“De los nodos intermedios hacia arriba, no se hace Copia”

Inserción de la clave 66



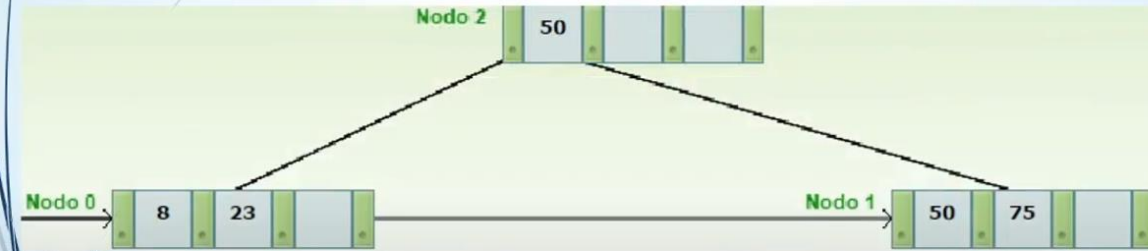
Insertion



9

+8

8 23 50 75
izq der



4 Orden
4 direcciones
3 elemts max
1 elemt min

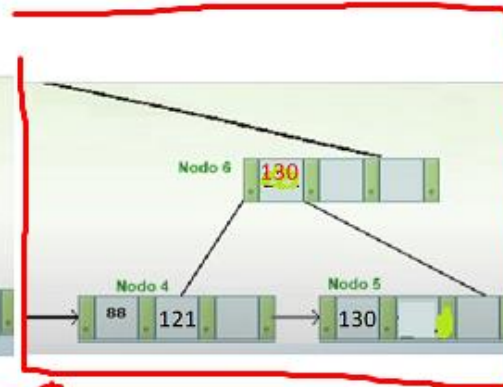
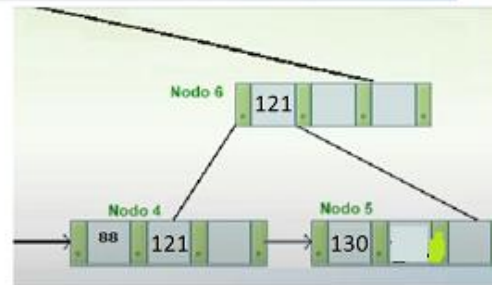
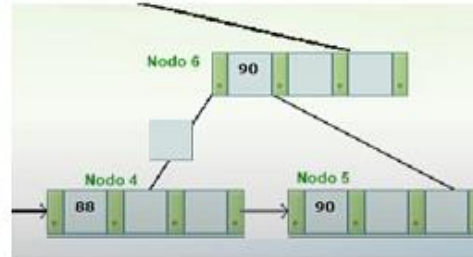
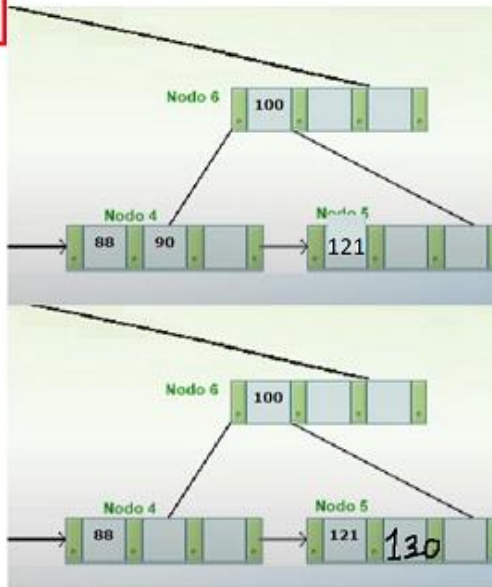
POLITICA, IZQUIERDA O DERECHA

DISTRIBUCIONES

A - 121
PEDIR IZQUIERDA

B

-88
PEDIR DERECHA



Bajas

Bajas en B+

La operación de eliminación en árboles B+ es más simple que en árboles B. Esto ocurre porque las claves a eliminar **siempre se encuentran 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 $\lceil M/2 \rceil - 1$, entonces termina la operación. Las claves de los nodos raíz o internos no se modifican por más que sean una copia de la clave eliminada en las hojas.

UnderFlow

Bajas en B+

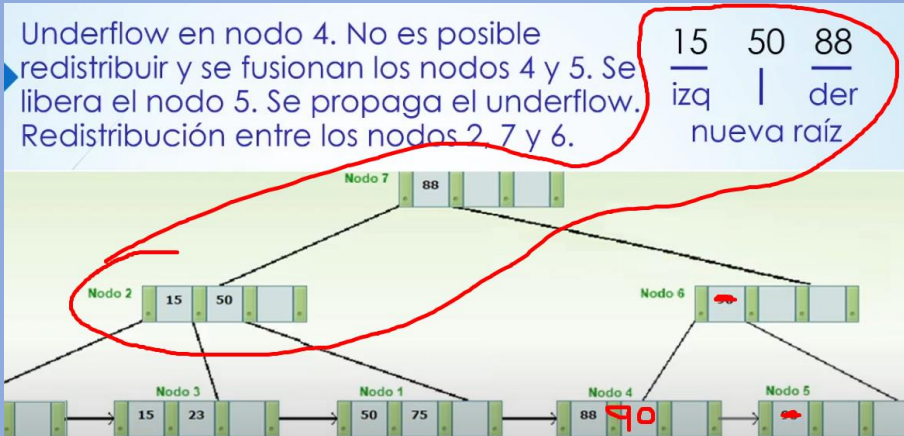
Underflow

- Si al eliminar una clave, la cantidad de llaves es menor a $[M/2]-1$, entonces debe realizarse una **redistribución** de claves, tanto en el índice como en las páginas hojas.
- Si la redistribución no es posible, entonces debe realizarse una **fusión** entre los nodos.

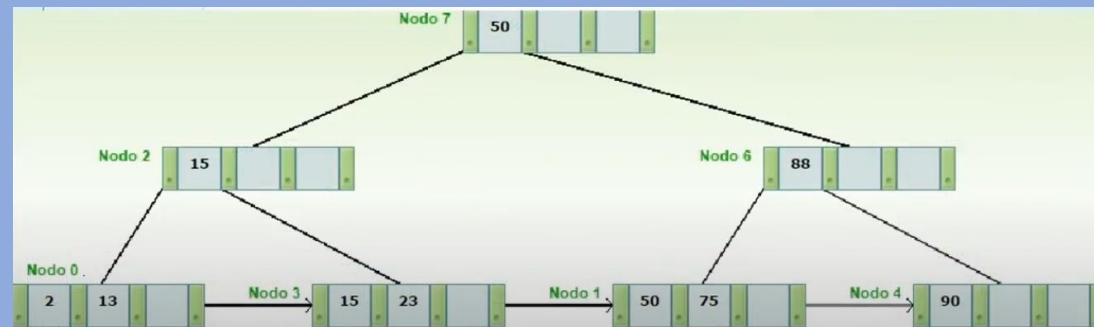
Políticas para la resolución de underflow:

1. **Política izquierda:** se intenta redistribuir con el hermano adyacente izquierdo, si no es posible, se fusiona con hermano adyacente izquierdo.
2. **Política derecha:** se intenta redistribuir con el hermano adyacente derecho, si no es posible, se fusiona con hermano adyacente derecho.
3. **Política izquierda o derecha:** se intenta redistribuir con el hermano adyacente izquierdo, si no es posible, se intenta con el hermano adyacente derecho, si tampoco es posible, se fusiona con hermano adyacente izquierdo.
4. **Política derecha o izquierda:** se intenta redistribuir con el hermano adyacente derecho, si no es posible, se intenta con el hermano adyacente izquierdo, si tampoco es posible, se fusiona con hermano adyacente derecho.

Cuando hay fusion se elimina al separador



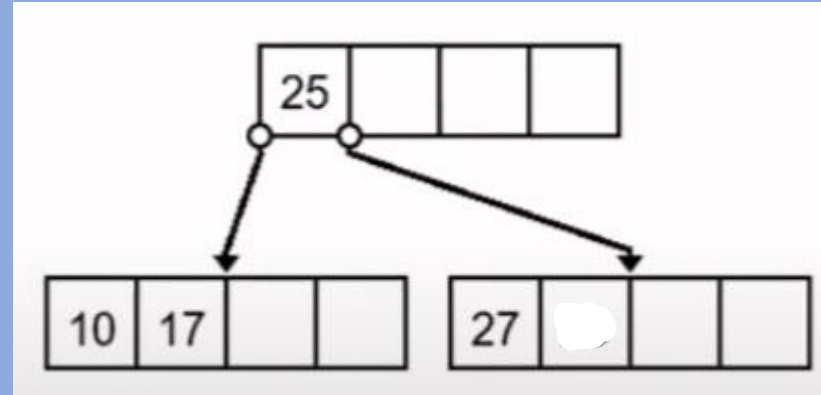
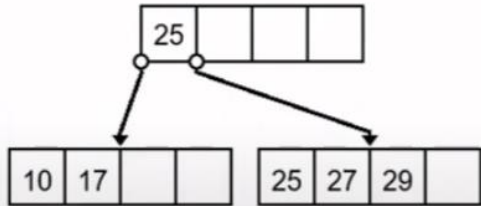
Tema fusion/Distribucion, si se hace un nodos no terminals se trata como un arbol B



Casos eliminacion

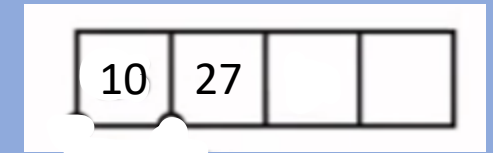
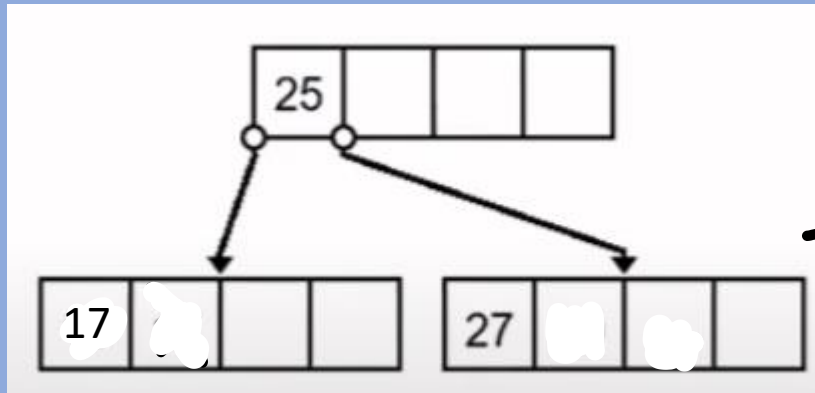
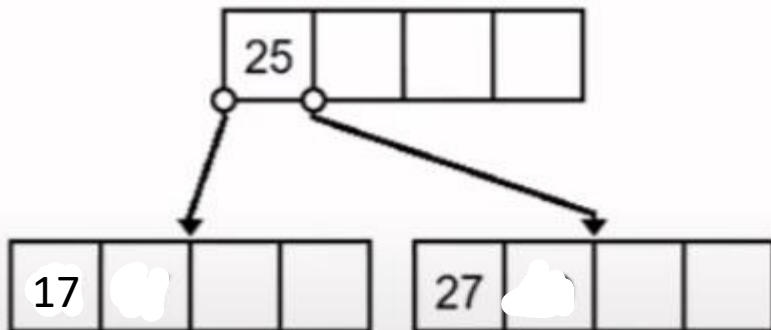
Mejor caso

- Ejemplo 1 Eliminar la clave 25



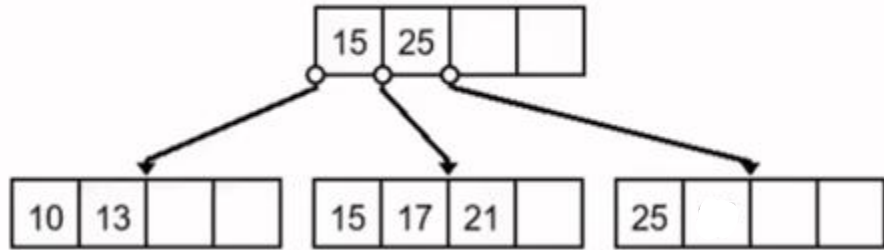
Se genera UnderFlow, se intenta redistribuir el su hermano ya tiene el minimo De claves, entonces se debe fusionar/concatenar.
Al fusionar se borra la referencia de arriba, 25.. El 25 no es clave, es referencia

Ejem 2, Eliminar la clave 17



Casos

- Ejemplo 2 Eliminar la clave 25



Se genera UnderFlow, se redistribuye, y el valor pasado 21, deja una copia en el separador

