

Fundamentos de Organización de Datos

Árboles B

Árboles B y B+

Los árboles B son árboles multicamino con una construcción especial de árboles que permite mantenerlos balanceados a bajo costo.

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.

Declaración de tipos de datos

const $M = \dots$; *{orden del árbol}*

type

nodo = **record**

cant_claves: **integer**;

claves: **array**[1.. $M-1$] of <tipo_dato>;

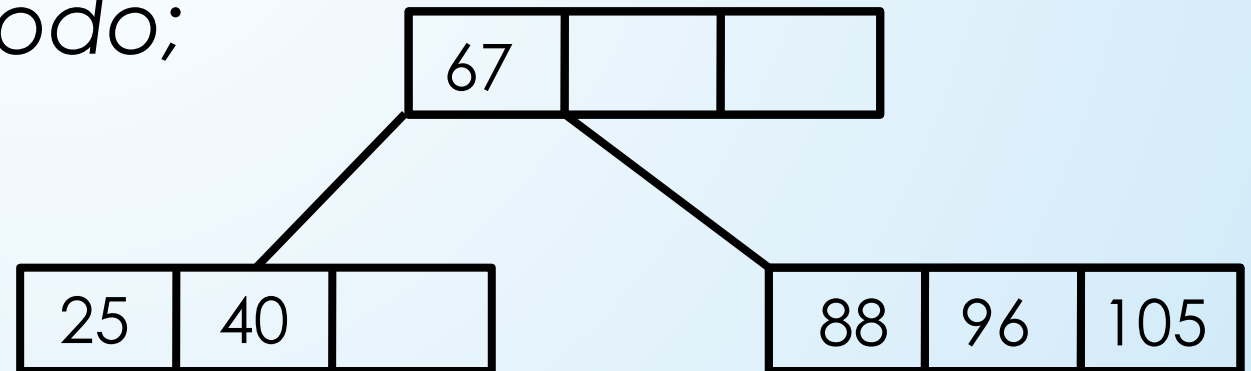
hijos: **array**[1.. M] of **integer**;

end;

arbol = **file of** nodo;

var

arbolB: arbol;



Ejemplo – Árbol B de orden 4

Árbol Inicial

Nodo 0

25	40	96
---------------	---------------	----

+40, +96, +25, +67

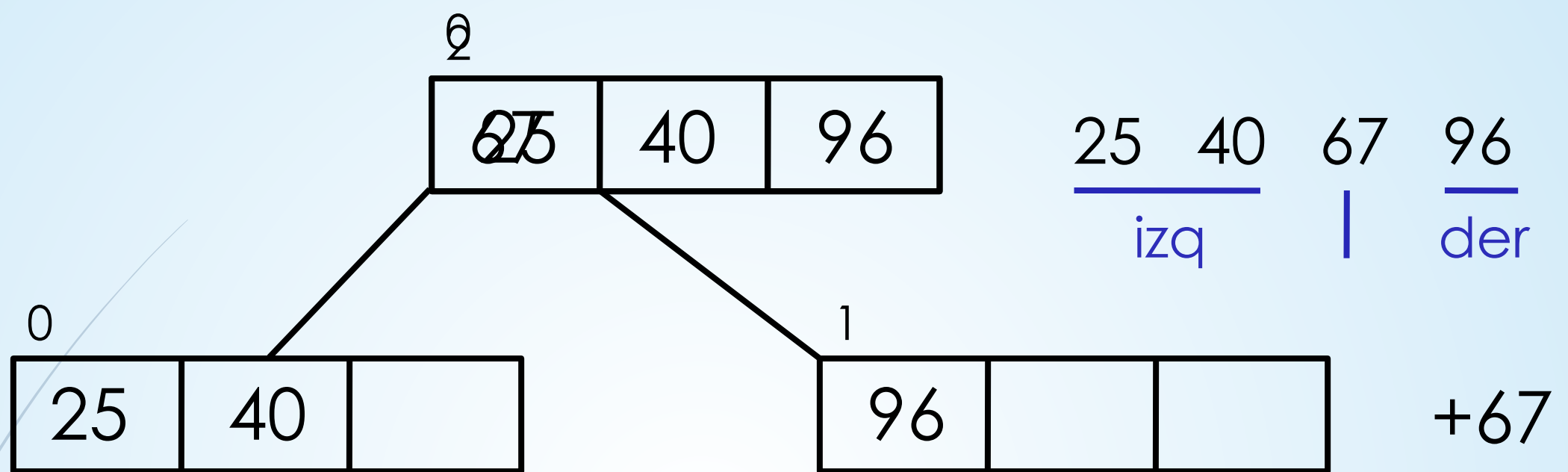
Archivo:

cant_claves: 0	25	40	96	
claves:	1	2	3	
hijos:	-1	-1	-1	-1
	1	2	3	4

NRR 0

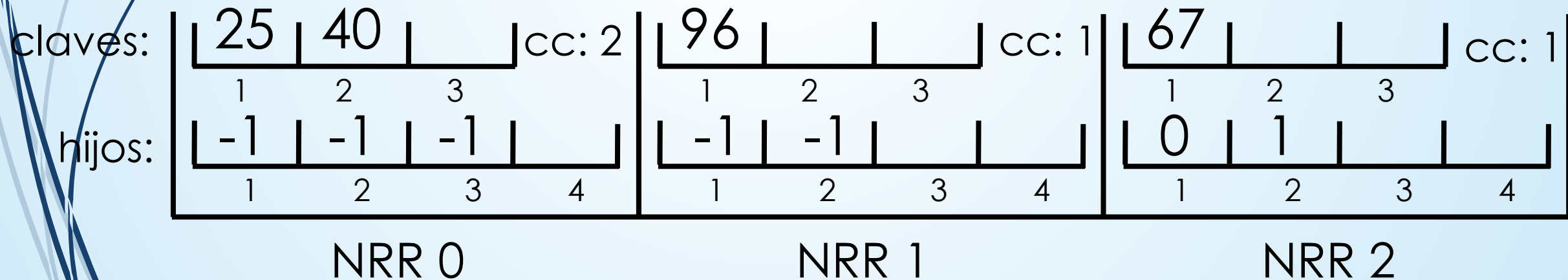
Overflow

- 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.



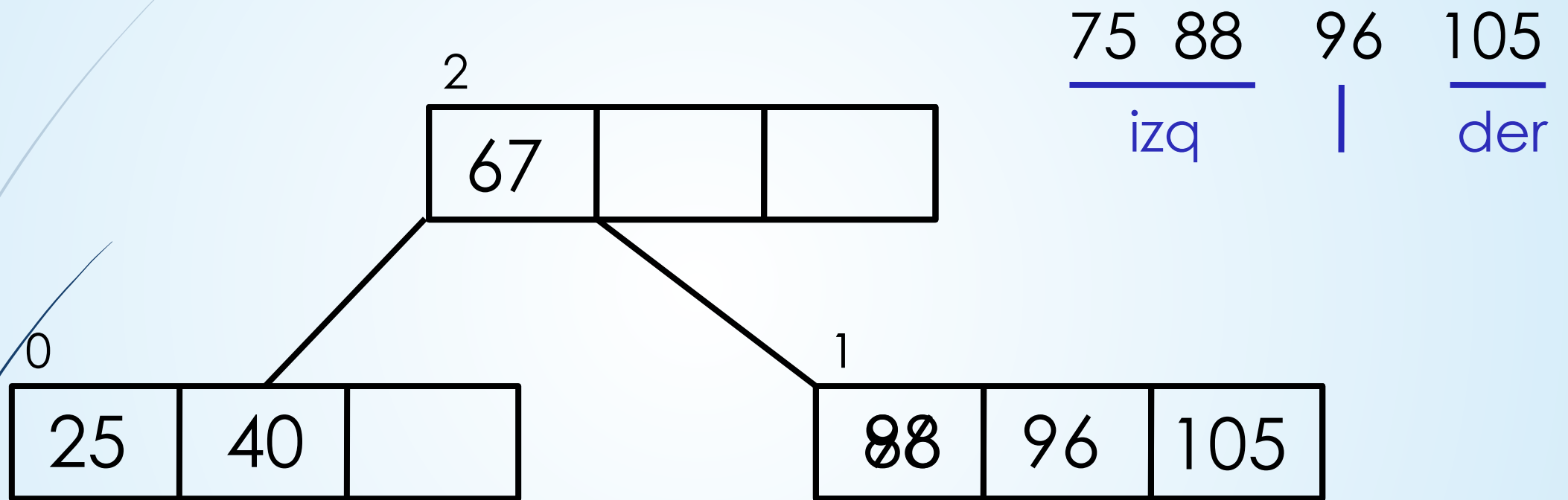
División de la raíz. Se incrementa la altura del árbol.

Archivo:



¡Notar la numeración de los nodos!

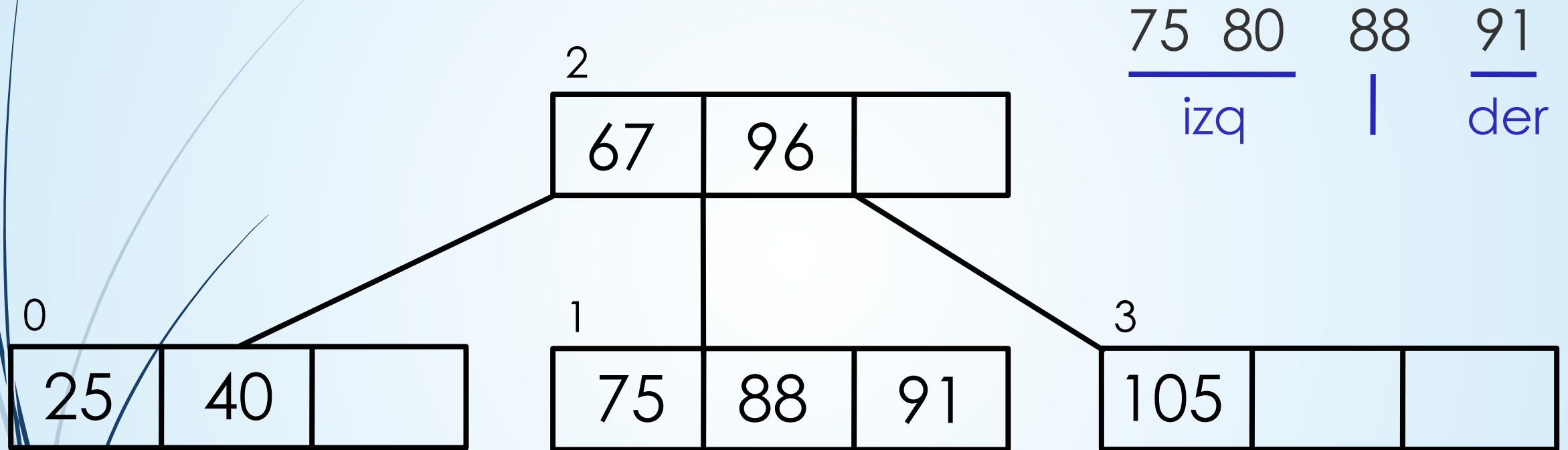
Ejemplo – Árbol B de orden 4



+88 , +105 , +75

Overflow en el nodo 1. División del mismo y promoción de la clave 96.

Ejemplo – Árbol B de orden 4

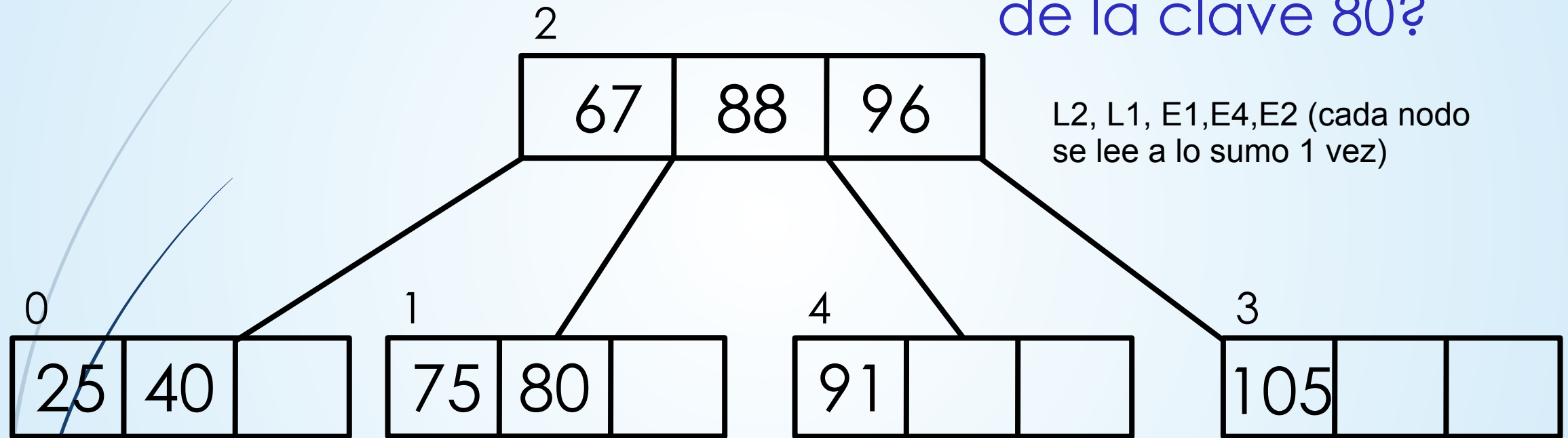


+75, +91, +80

Overflow en el nodo 1. División del mismo y promoción de la clave 88.

Ejemplo – Árbol B de orden 4

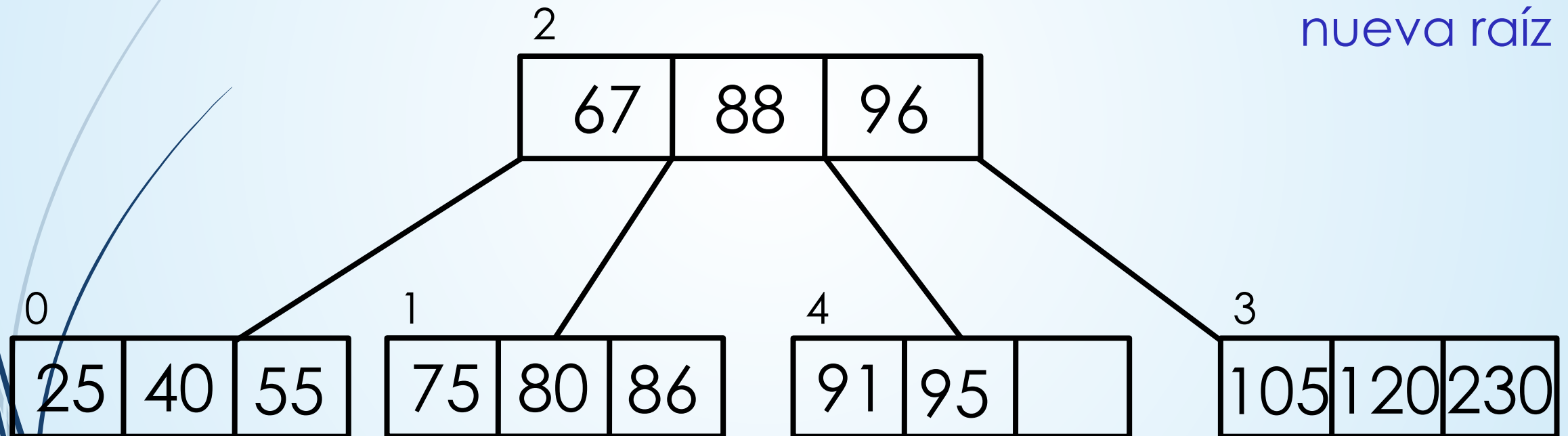
¿L/E necesarias para el alta de la clave 80?



+80

Overflow en el nodo 1. División del mismo y promoción de la clave 80.

Propagación del overflow a la raíz. División de la misma y aumento en la altura del árbol.



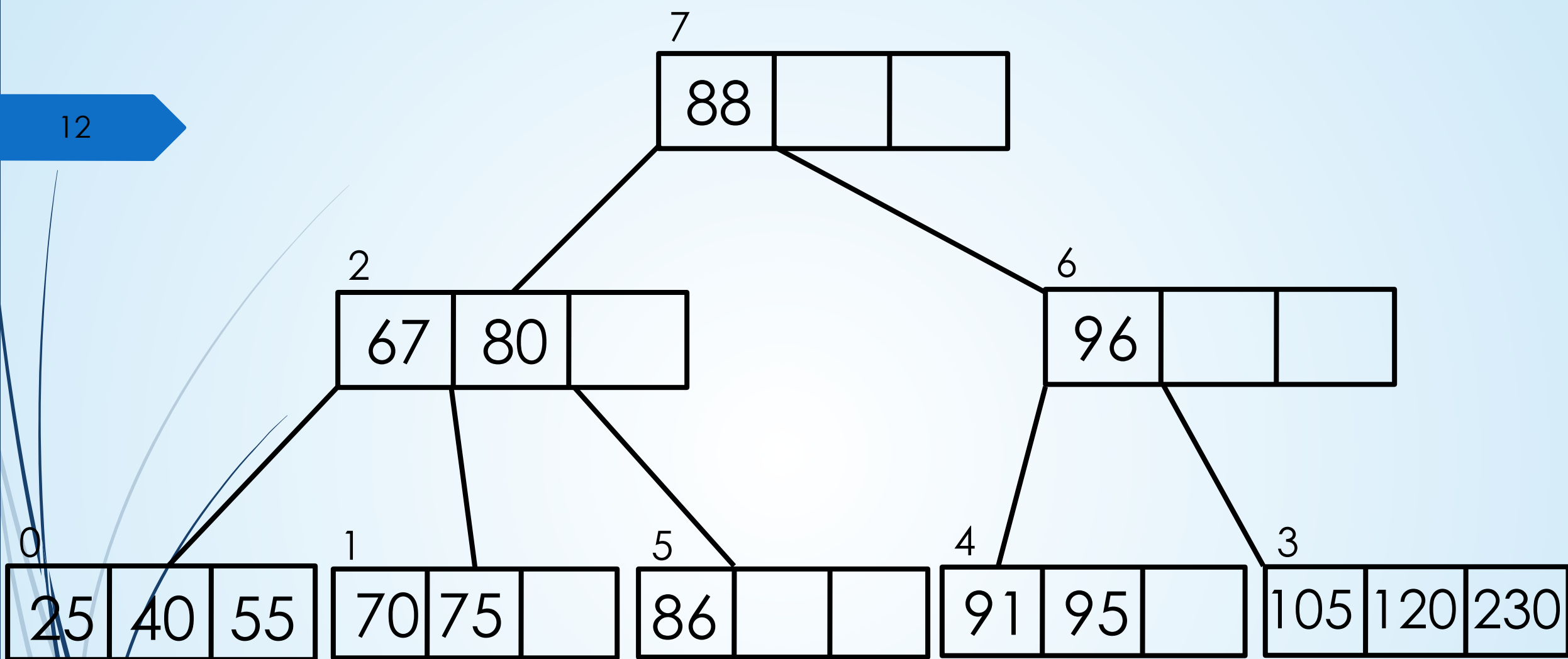
+86 , +120 , +230 , +95 , +55 , +70

70	75	80	86
<hr/>			<hr/>
izq			der

67	80	88	96
<hr/>			<hr/>
izq			der

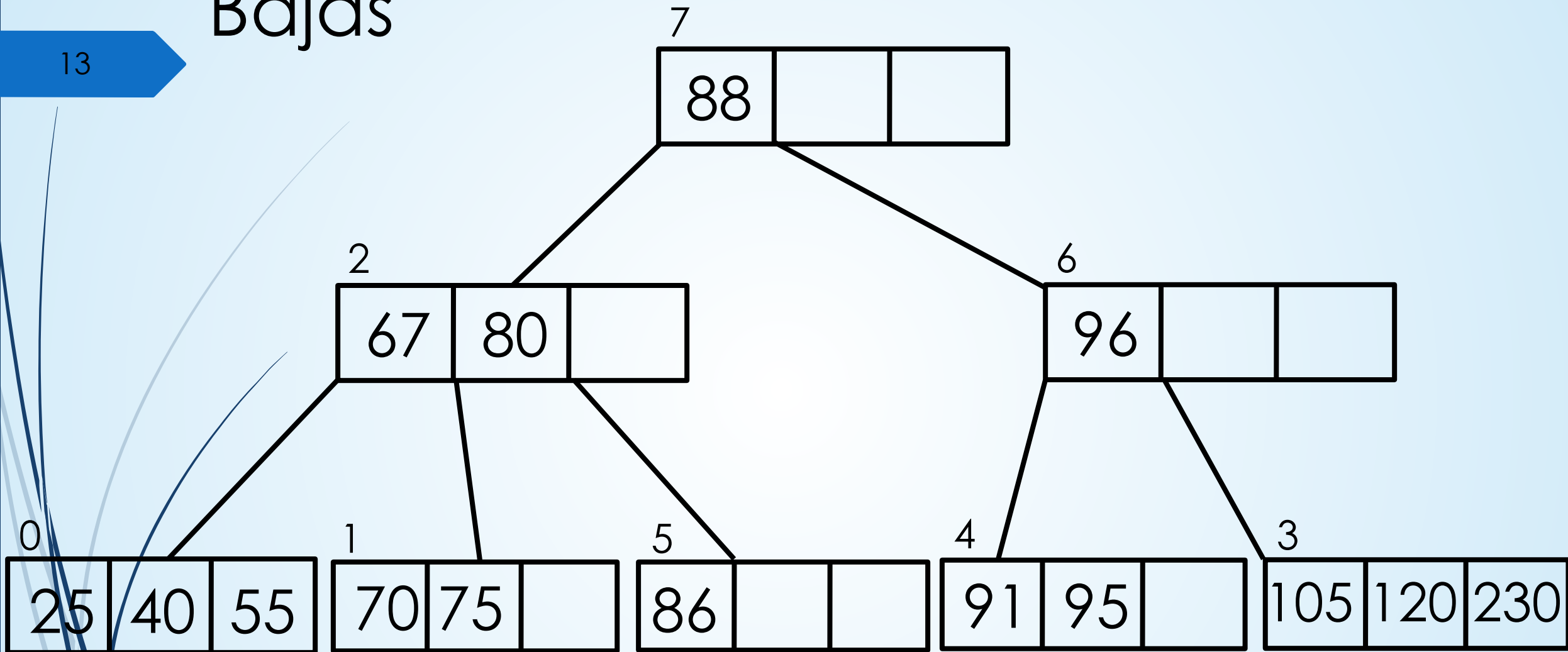
nueva raíz

12



Bajas

13



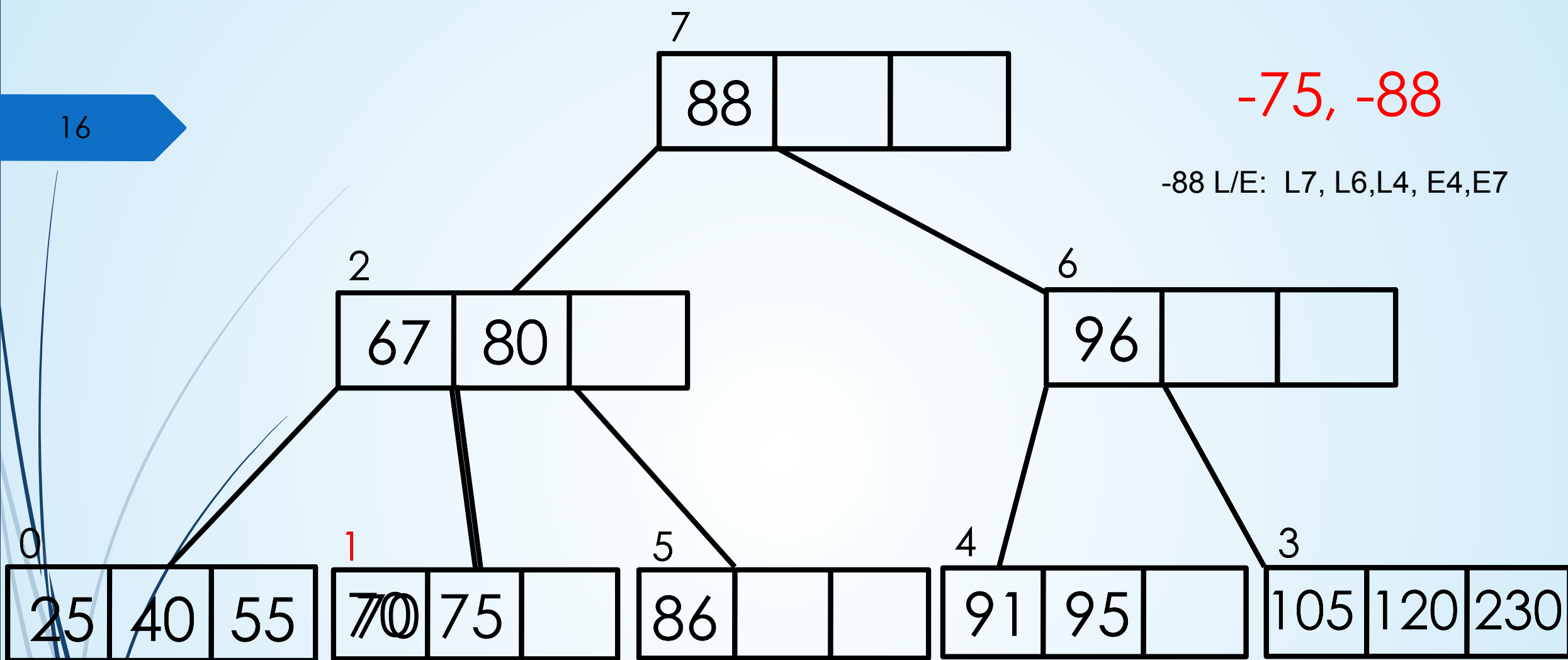
-75

Bajas

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. En caso contrario, se debe tratar el underflow

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.

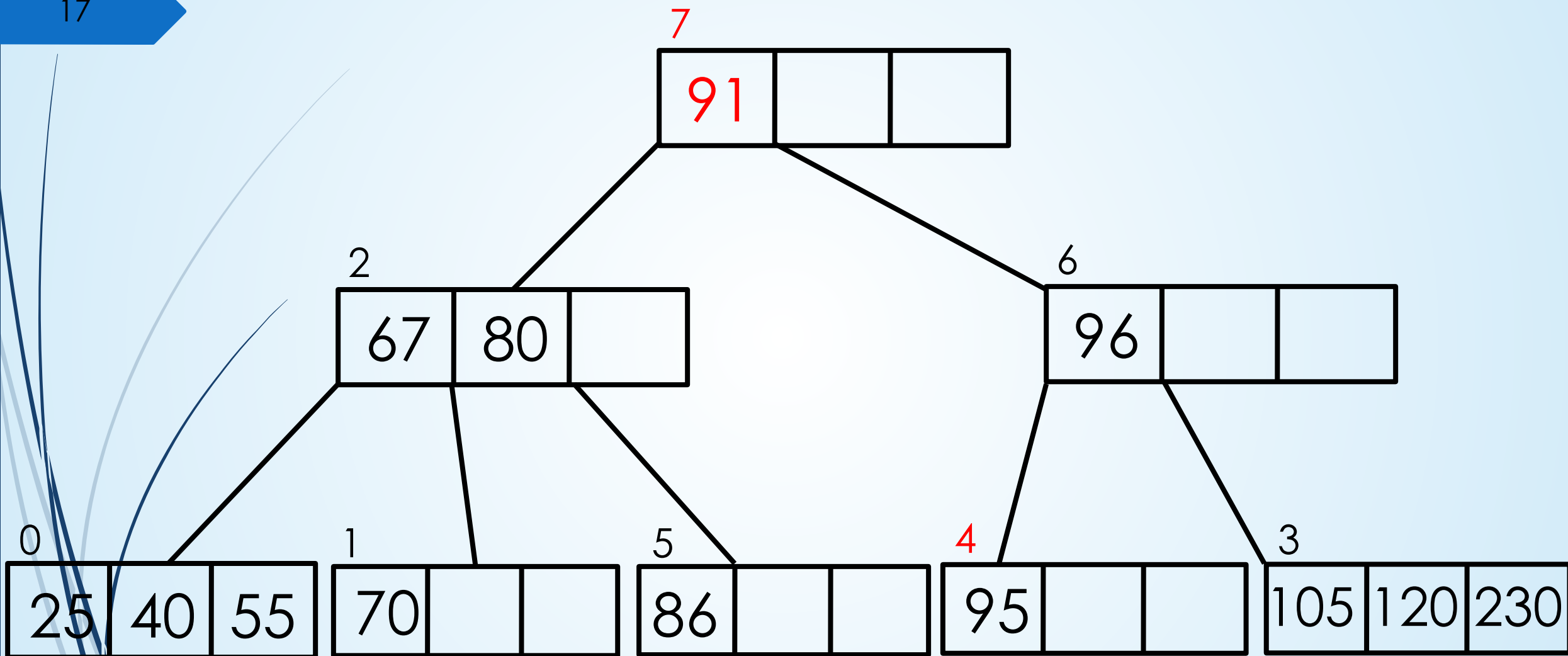


Eliminación de la clave 75 en el nodo 1.

Baja del 88, se reemplaza la clave por la menor clave del subárbol derecho.

No se genera underflow en la hoja

Ejemplo



-88 , -70

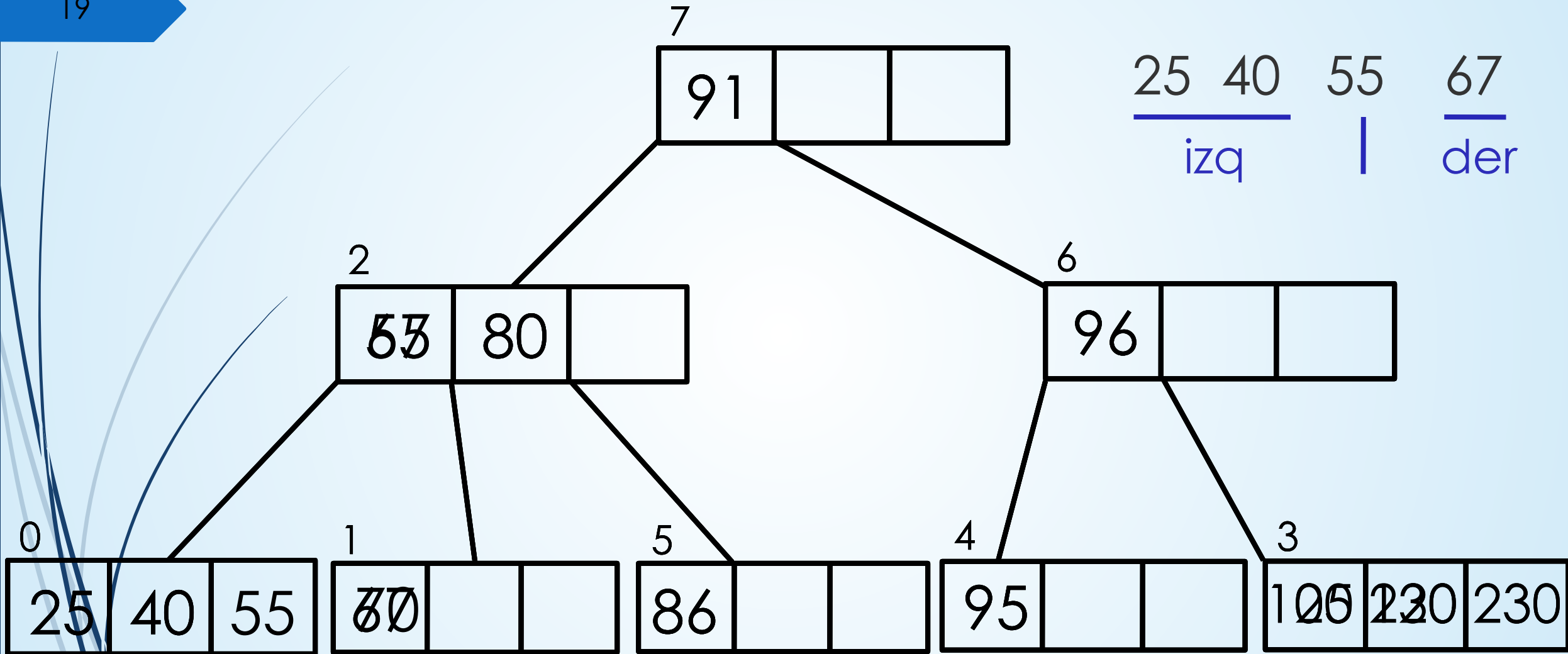
Baja de la clave 70

La eliminación de la clave 70 en el nodo 1 produce underflow.

Se intenta redistribuir con el hermano derecho. No es posible ya que el nodo contiene la cantidad mínima de claves.

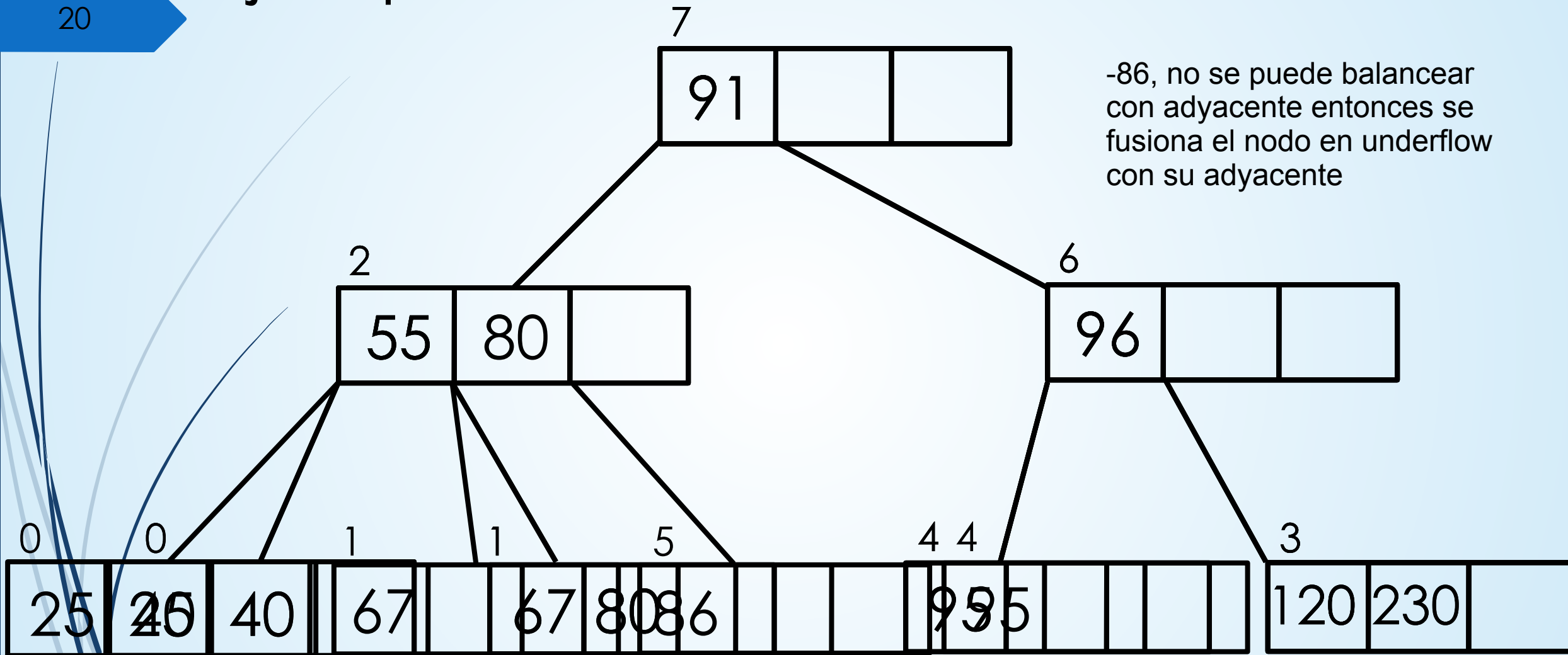
Se intenta redistribuir con el hermano izquierdo. La operación es posible y se rebalancea la carga entre los nodos 1 y 0.

Ejemplo



-70 , -105 , -86

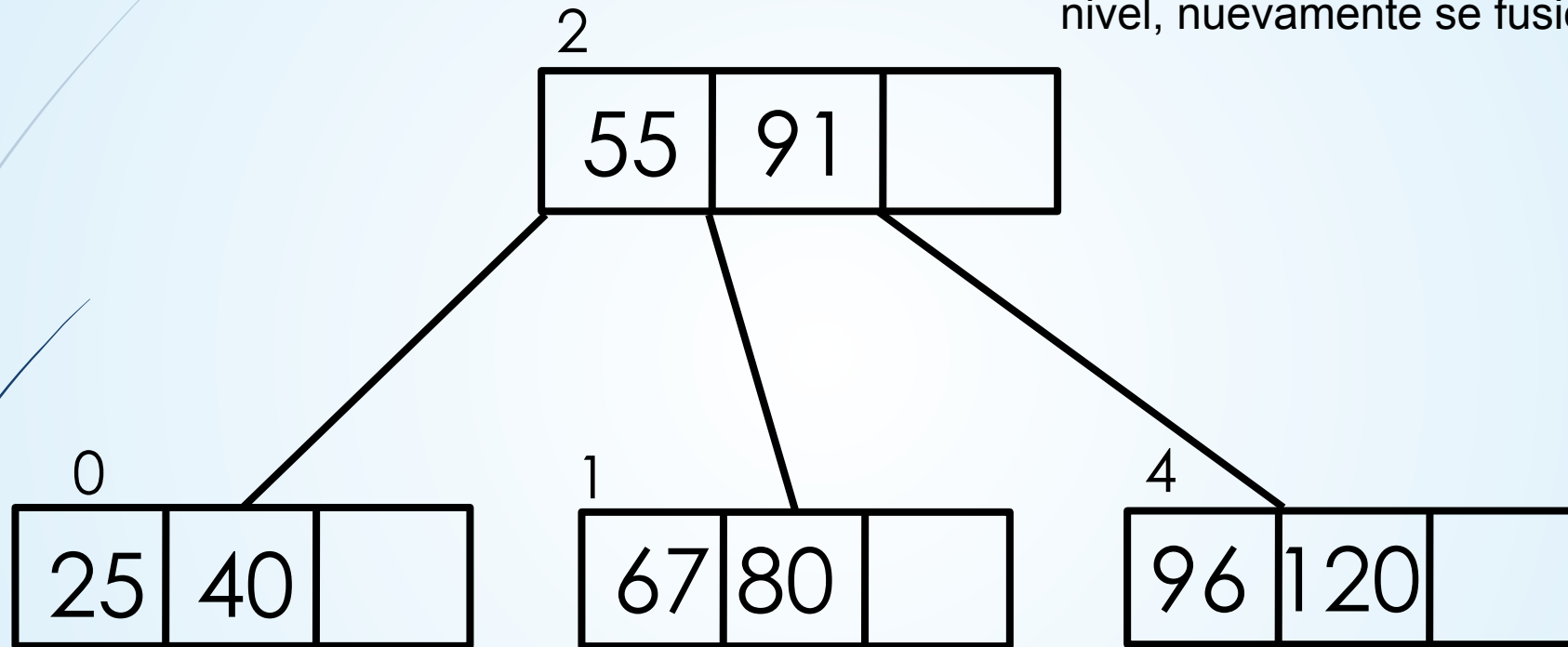
Ejemplo



-86, -230, -95

Ejemplo

-95, no se puede balancear con adyacente entonces se fusiona el nodo en underflow con su adyacente, generando underflow en el siguiente nivel, nuevamente se fusiona



-95

Ej: Redistribución en nodo interno

