

ESTRUCTURA DE DATOS 2

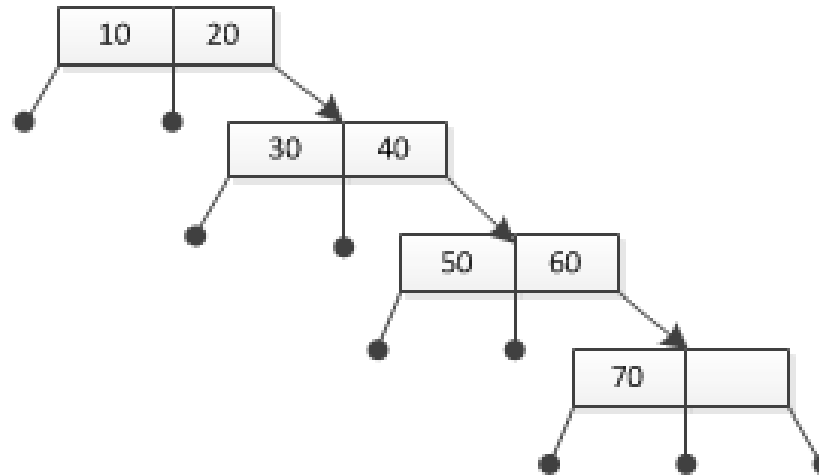
ARBOLES B

ARBOLES B

- Problema de los arboles
 - Degeneración
 - Desbalanceo exagerado
 - En el caso de un árbol binario, cuando el desbalanceo es exagerado, lo que obtenemos es una lista.
 - Los arboles m vías, al tener $(m - 1)$ datas por nodo, tardan más en desbalancearse, pero también lo hacen.

EJEMPLO DEL PROBLEMA

- Insertando 10,20,30,40,50,60 y 70, en un arbol de 3 vías:



ARBOLES M-VIAS BALANCEADOS (ARBOLES B)

- El uso de arboles de m-vías con el tamaño de nodo igual al bloque de lectura del disco, permite reducir el tiempo de acceso.
- En un solo acceso se recuperan (m-1) datas.
- Recordar, los arboles m-vías se pueden desbalancear.
- Los arboles B, son arboles de m-vías balanceados.

ARBOLES B

- PROPIEDADES DE UN ARBOL DE M-VIAS
 - Si la raíz no es hoja, tiene como mínimo dos subárboles.
 - Cada nodo no raíz y no hoja contiene $k - 1$ datos y k referencias a los subárboles donde $(m/2) \leq k \leq m$
 - Cada nodo hoja contiene $k - 1$ datos donde $(m/2) \leq k \leq m$
 - Todas las hojas están en el mismo nivel.
- Mínimos y Máximos de datos e Hijos por nodo:
 - $\text{nroMaximoDeHijos} = \text{orden}$
 - $\text{nroMaximoDeDatos} = \text{orden} - 1$
 - $\text{nroMinimoDeDatos} = \text{nroMaximoDeDatos} / 2$ (División entera)
 - $\text{nroMinimoDeHijos} = \text{nroMinimoDeDatos} + 1$

ARBOLES B – Resumen de Reglas

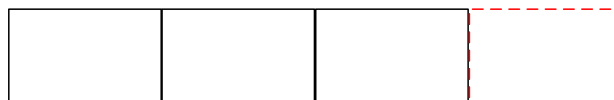
- Todo nodo hoja debe tener al menos el mínimo de datos. La raíz es la excepción, si es hoja puede tener por debajo del mínimo de datos.
- Todo nodo no hoja debe tener el mínimo de datos y el mínimo de hijos. La raíz es la excepción, si no es nodo hoja puede romper la regla del mínimo de datos y el mínimo de hijos, pero debe tener al menos 2 hijos.
- Ningún nodo hoja puede tener más del máximo de datos, ni la raíz.
- Ningún nodo no hoja puede romper la regla del máximo de datos ni el máximo de hijos.

ARBOLES B - INSERCIÓN

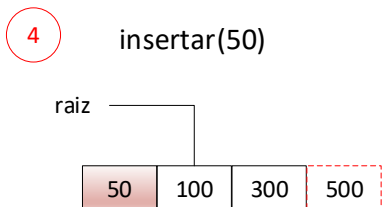
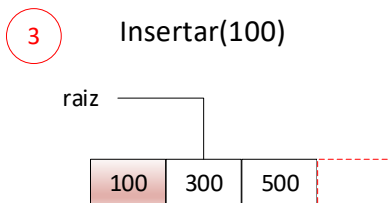
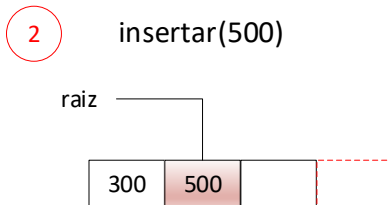
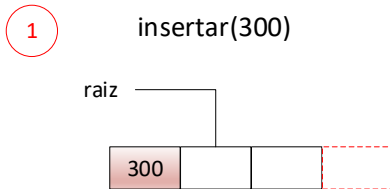
- El árbol crece hacia arriba.
- Deja muchas datas vacías (No le importa)
- Cada elemento busca una hoja para insertarse
 - Si la hoja donde debe insertarse el nuevo dato hace que sobrepase el máximo de datos permitidos por nodo, este nodo hoja se debe dividir en dos, para lo cual se ubica el valor medio de los datos que hay en el nodo incluido el dato a insertar.
 - Los valores a la izquierda de este valor, forman un nuevo nodo y los de la derecha forman otro.
 - El valor medio que se usa como separador irá a ubicarse en el nodo padre del nodo hoja, aumentando la cantidad de datos del padre y como el nodo hoja se dividió en dos, también se incrementan el número de hijos del nodo padre.
 - Cuando se inserta el valor medio en un nodo padre puede ocurrir el mismo efecto que cuando insertamos un nuevo dato en una hoja, provocando que el padre tenga que dividirse por un valor medio y así sucesivamente

ARBOLES B – INSERCIÓN EN ÁRBOL DE ORDEN 4

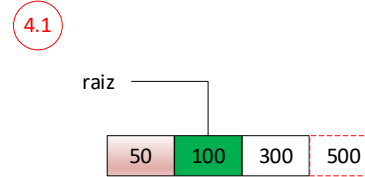
- $\text{orden} = \text{nroMaximoDeHijo} = 4$
- $\text{nroMaximoDeDatos} = 3$
- $\text{nroMinimoDeHijos} = 2$
- $\text{nroMinimoDeDatos} = 1$
- El nodo para nuestro árbol lo utilizaremos con una posición adicional para datos e hijos, que solo se utilizará en forma temporal:



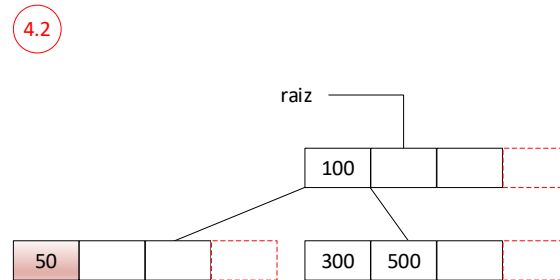
INSERCIÓN – P.1.



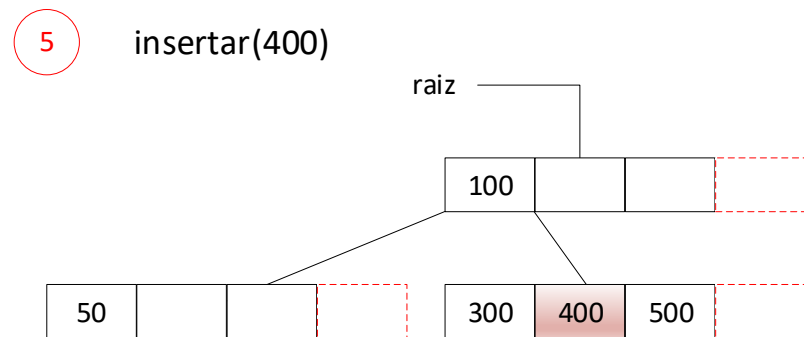
Se pasa del
máximo de
datos



Encontramos valor
medio. Es 100. Debe
subir al nodo padre.



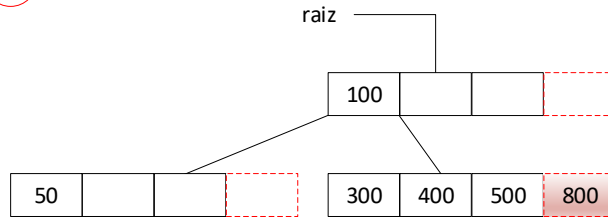
Como el nodo no tiene
padre, se crea un nuevo
nodo, que será el padre, a
donde subirá el valor 100 y
se convierte en el nuevo
nodo raiz



INSERCIÓN – P.2.

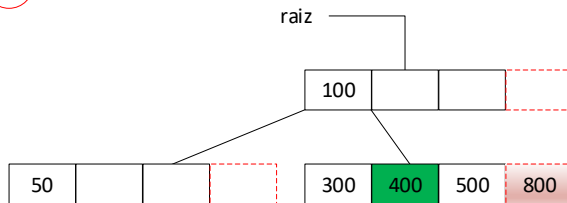
6

insertar(800)



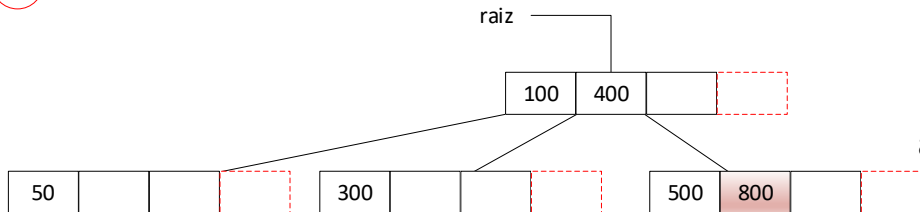
Se pasa del
máximo de
datos

6.1



Encontramos el valor
medio. Es 400. Debe
subir al nodo padre.

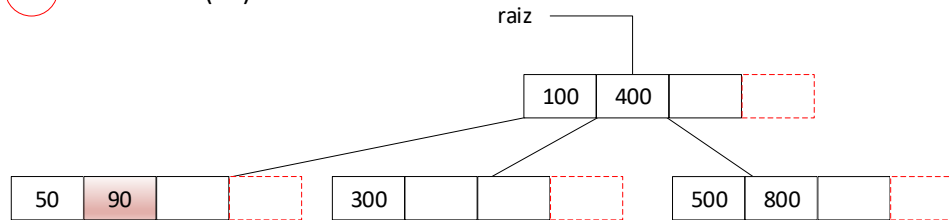
6.2



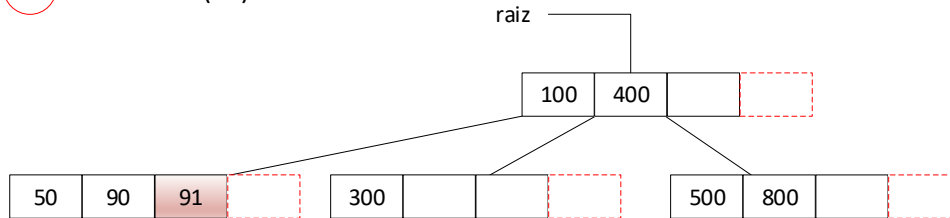
El 400 sube al nodo padre. El
nodo hoja se divide en dos
nodos y el padre ahora tiene
un dato más y un hijo
adicional. El padre no rompe
ninguna regla

INSERCIÓN – P.3.

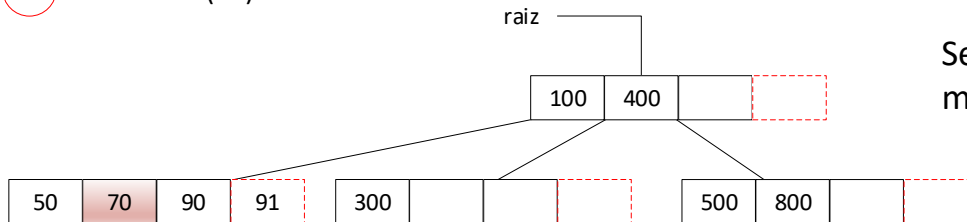
7 insertar(90)



8 insertar(91)



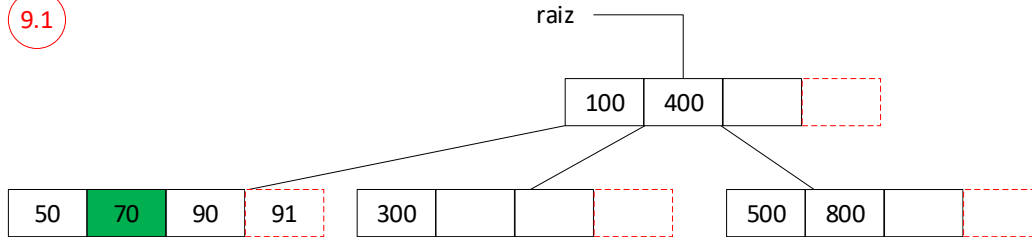
9 insertar(70)



Se pasa del
máximo de
datos

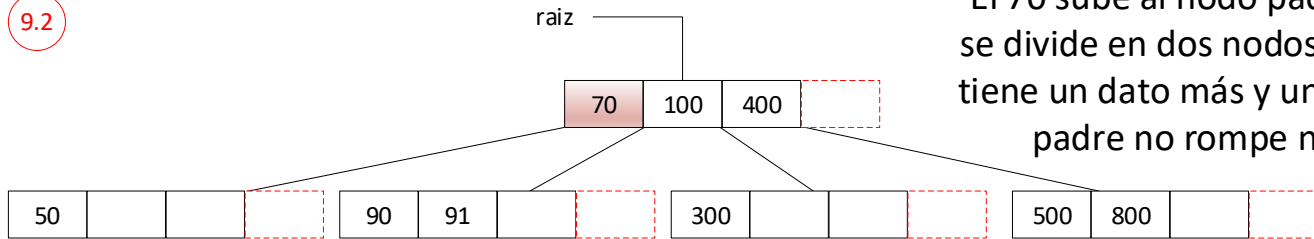
INSERCIÓN – P.4.

9.1



Encontramos el valor medio. Es 70. Debe subir al nodo padre.

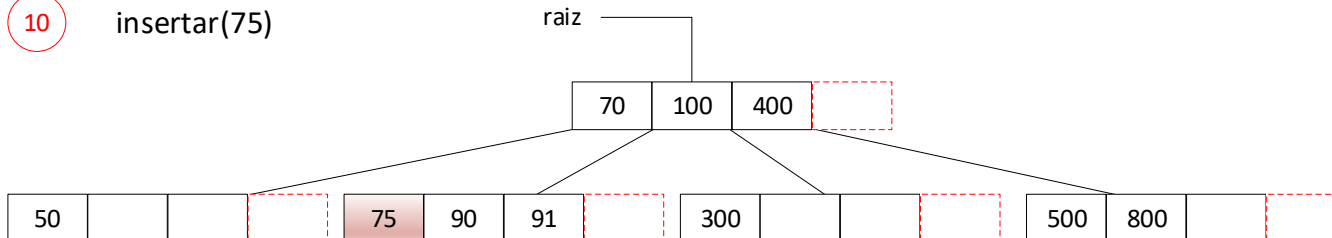
9.2



El 70 sube al nodo padre. El nodo hoja se divide en dos nodos y el padre ahora tiene un dato más y un hijo adicional. El padre no rompe ninguna regla

10

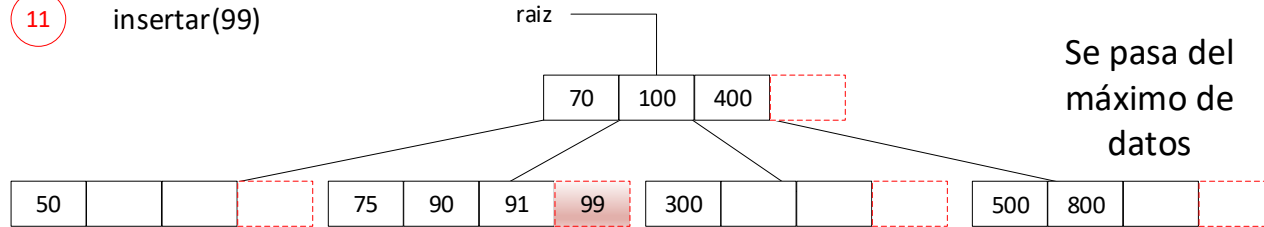
insertar(75)



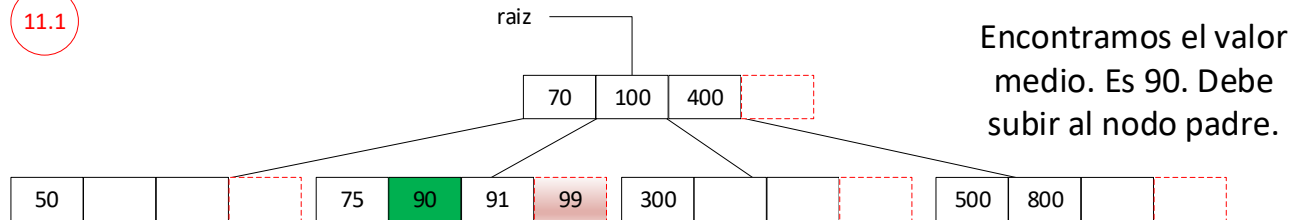
INSERCIÓN – P.5.

11

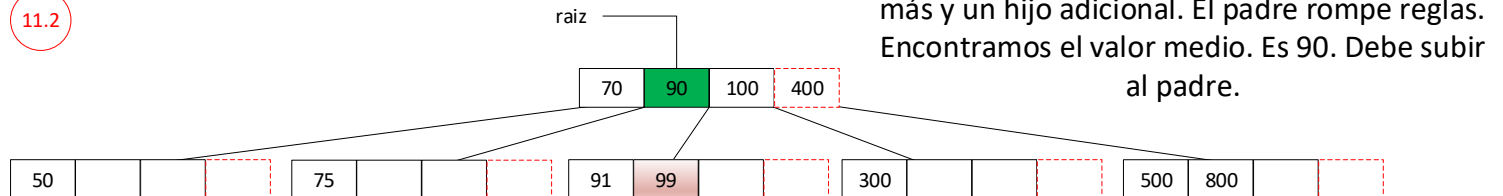
insertar(99)



11.1

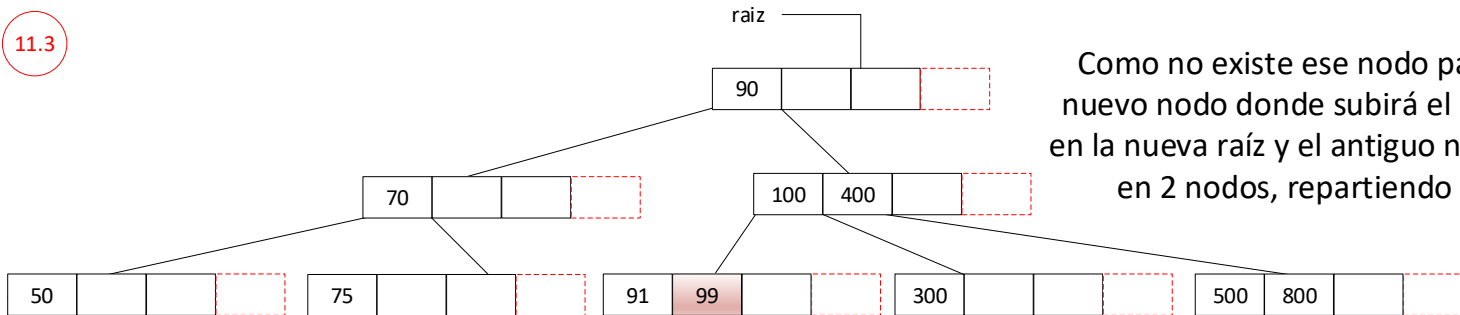


11.2



INSERCIÓN – P.6.

11.3



Como no existe ese nodo padre, se crea un nuevo nodo donde subirá el 90, se convertirá en la nueva raíz y el antiguo nodo raíz se divide en 2 nodos, repartiendo datos e hijos

ARBOL - ELIMINACIÓN

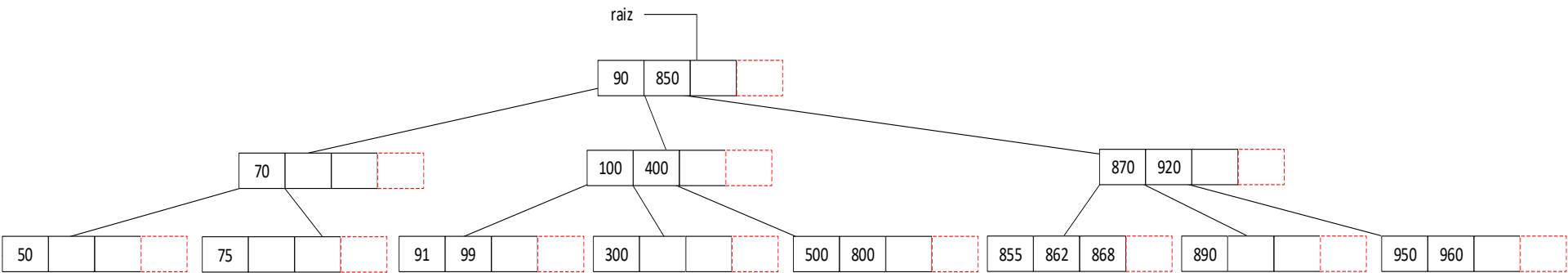
- Si el dato a eliminar cae en una hoja y al eliminar dicho dato, el nodo hoja no rompe el mínimo de datos, no se hace ninguna operación adicional.
- Si el dato a eliminar cae en una hoja y al eliminar dicho dato, el nodo hoja rompe el mínimo de datos, se tiene las opciones:
 - Prestarse un dato del hermano que está en la siguiente posición (si hay este hermano), pero solo si al prestar ese nodo no rompe ninguna regla. Nos presta su primer dato
 - Prestarse un dato del hermano que está en la anterior posición (si hay este hermano), pero solo si al prestar ese nodo no rompe ninguna regla. Nos presta su ultimo dato.
 - Si no se puede las dos primeras opciones, se puede fusionar con el hermano de la siguiente posición (si hay este hermano).
 - Si no se puede ninguna de las 3 opciones, solo queda fusionarse con el hermano de la anterior posición.
 - **IMPORTANTE.** Al realizar operaciones de fusión el nodo padre perderá un dato y un hijo. Si esto provoca que el padre rompa reglas, entonces con este nodo padre se vuelve a operar siguiendo cualquiera de estas 4 opciones, según se pueda y así sucesivamente.

ARBOL – ELIMINACIÓN (Cont.)

- Si el dato a eliminar en un nodo no hoja, lo que hacemos es buscarle un reemplazo.
 - El reemplazo será el predecesor inOrden
 - El reemplazo siempre cae en una hoja.
 - Antes de usar el reemplazo se lo debe eliminar de su posición original, lo cual puede desencadenar cualquiera de las situaciones expuestas en la anterior lamina.

ELIMINACIÓN EN ARBOL B - ORDEN 4

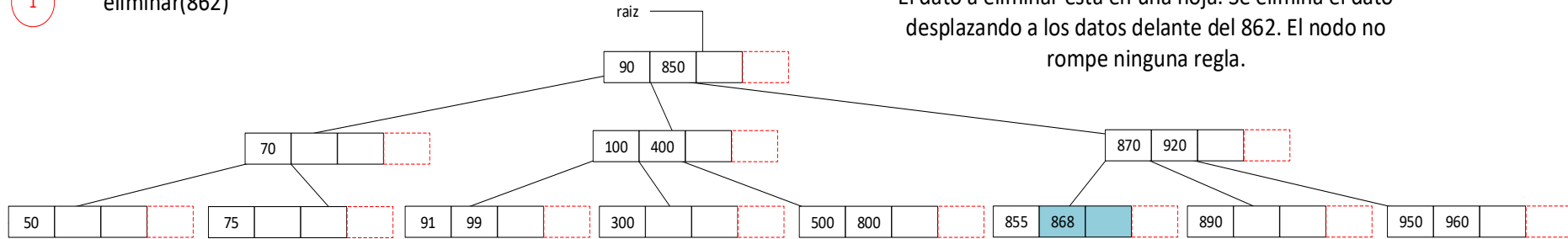
- orden = nroMaximoDeHijo = 4
- nroMaximoDeDatos = 3
- nroMinimoDeDatos = 1
- nroMinimoDeHijos = 2



ELIMINAR – P.1.

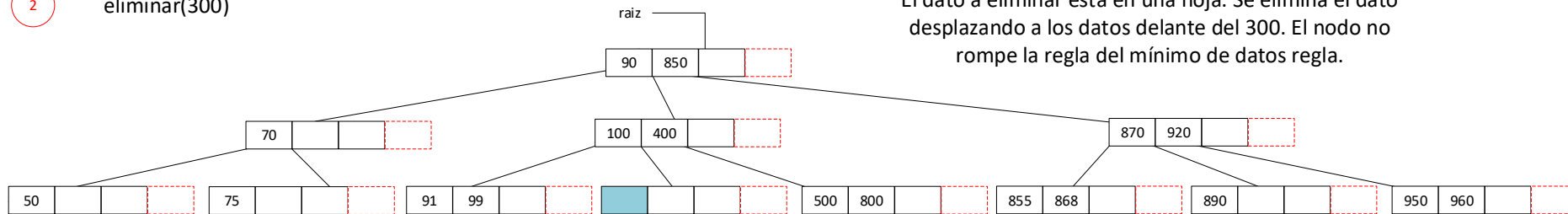
1

eliminar(862)



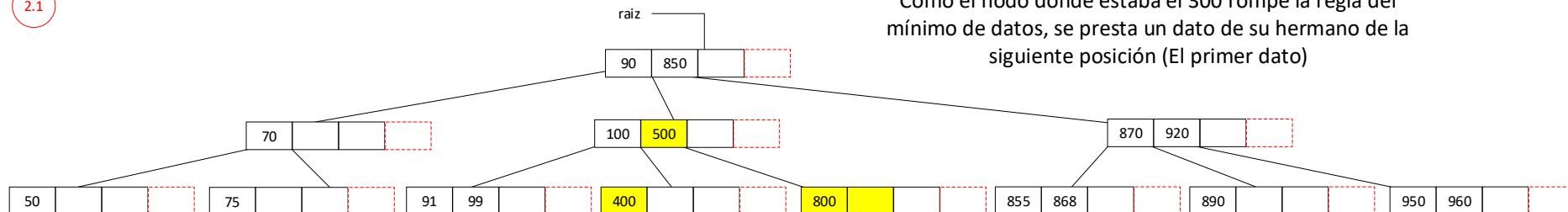
2

eliminar(300)



2.1

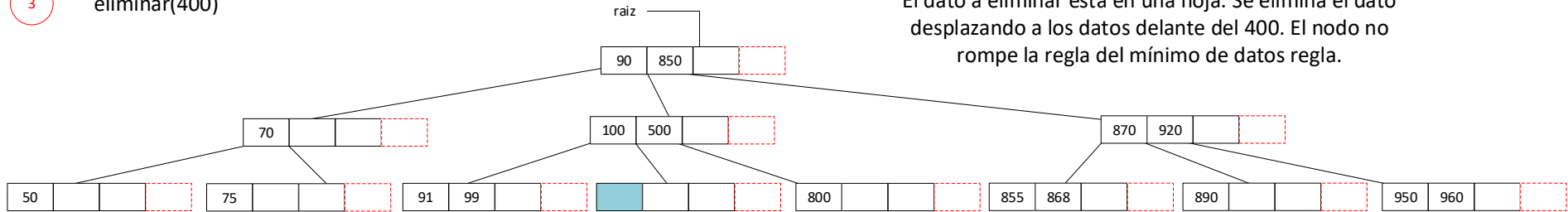
Como el nodo donde estaba el 300 rompe la regla del mínimo de datos, se presta un dato de su hermano de la siguiente posición (El primer dato)



ELIMINAR – P.2.

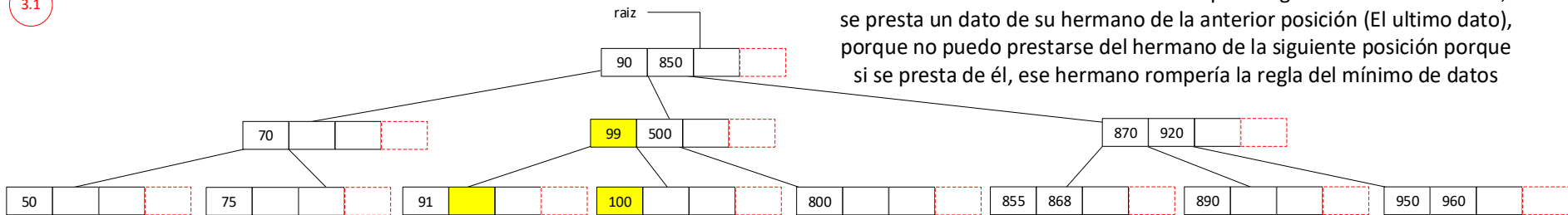
3

eliminar(400)



El dato a eliminar esta en una hoja. Se elimina el dato desplazando a los datos delante del 400. El nodo no rompe la regla del mínimo de datos regla.

3.1

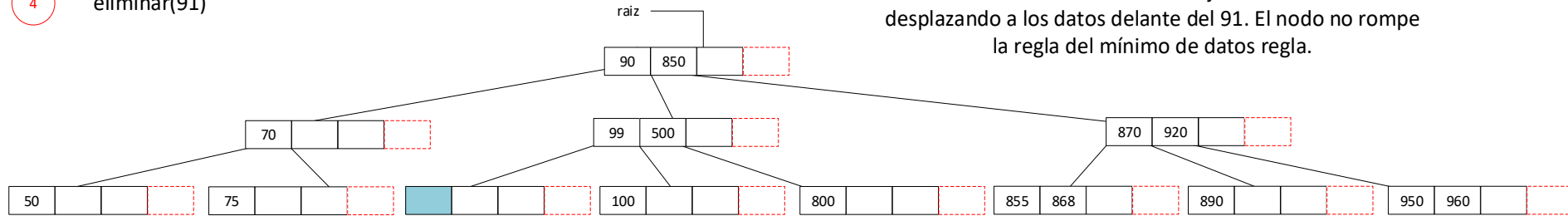


Como el nodo donde estaba el 400 rompe la regla del mínimo de datos, se presta un dato de su hermano de la anterior posición (El ultimo dato), porque no puedo prestarse del hermano de la siguiente posición porque si se presta de él, ese hermano rompería la regla del mínimo de datos

ELIMINAR – P.3.

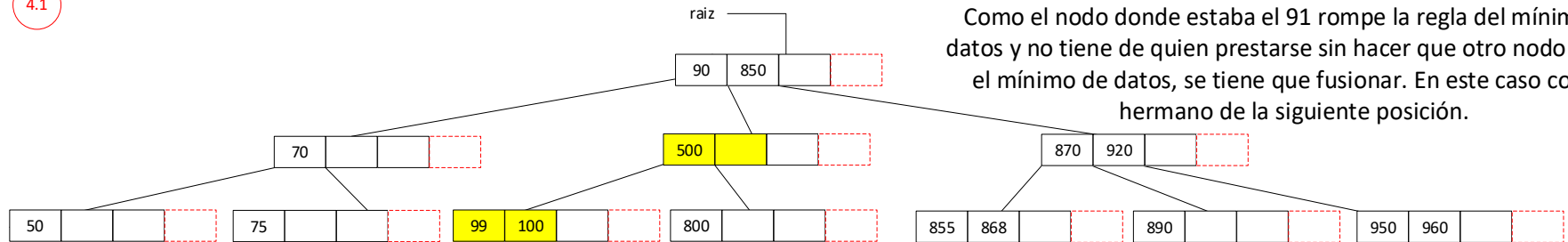
4

eliminar(91)



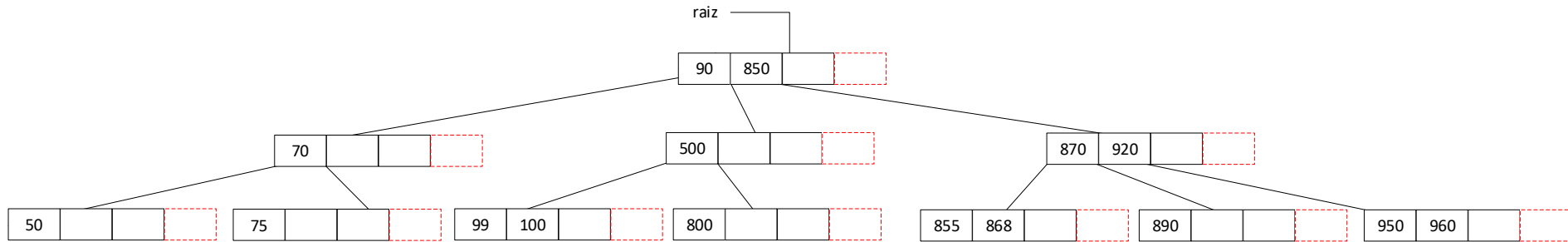
El dato a eliminar esta en una hoja. Se elimina el dato desplazando a los datos delante del 91. El nodo no rompe la regla del mínimo de datos regla.

4.1



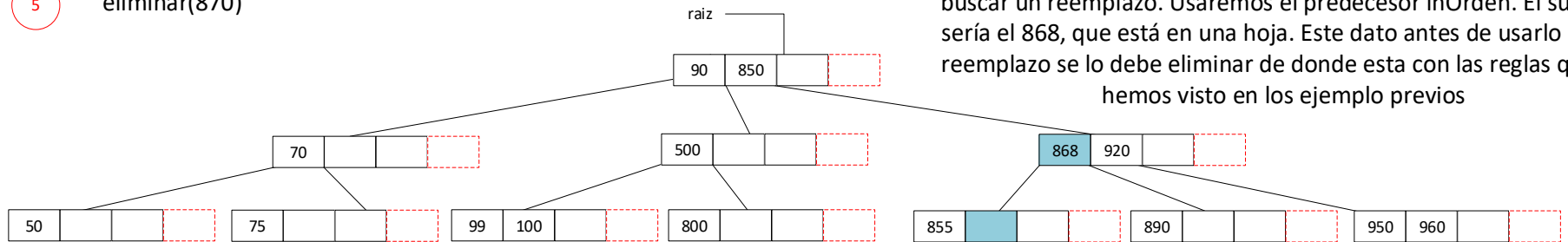
Como el nodo donde estaba el 91 rompe la regla del mínimo de datos y no tiene de quien prestarse sin hacer que otro nodo rompa el mínimo de datos, se tiene que fusionar. En este caso con su hermano de la siguiente posición.

ELIMINAR – P.4.



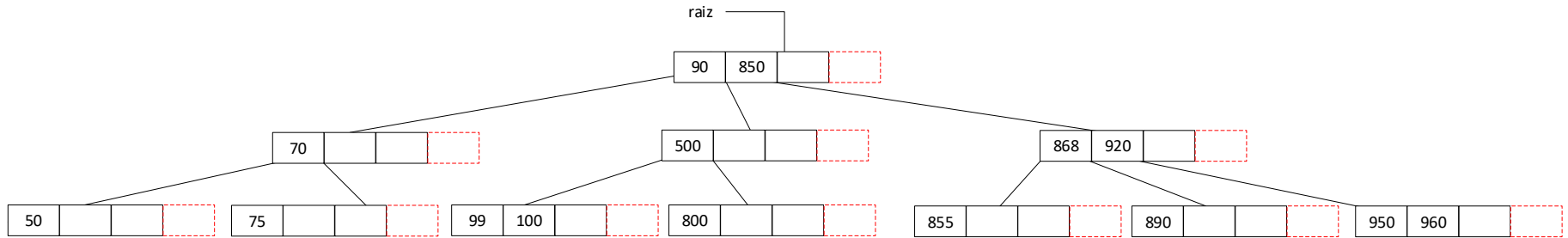
5

eliminar(870)



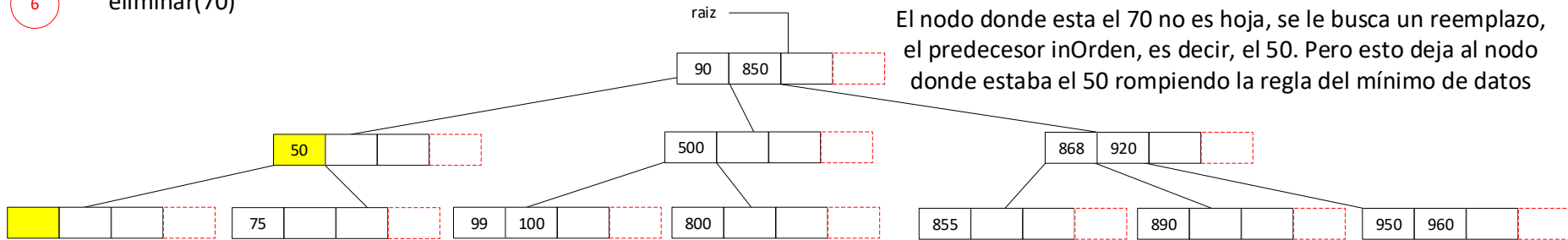
El nodo donde esta el dato no es una hoja, entonces se le debe buscar un reemplazo. Usaremos el predecesor inOrden. El sucesor sería el 868, que está en una hoja. Este dato antes de usarlo como reemplazo se lo debe eliminar de donde esta con las reglas que ya hemos visto en los ejemplo previos

ELIMINAR – P.5.



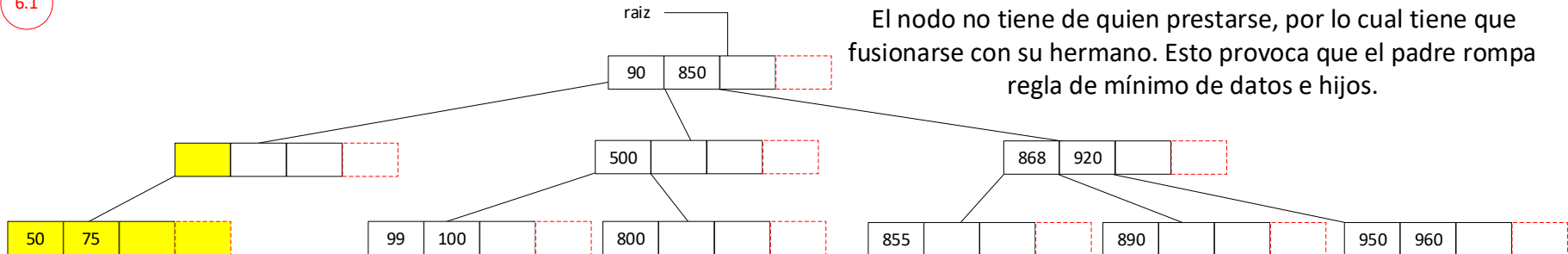
6

eliminar(70)



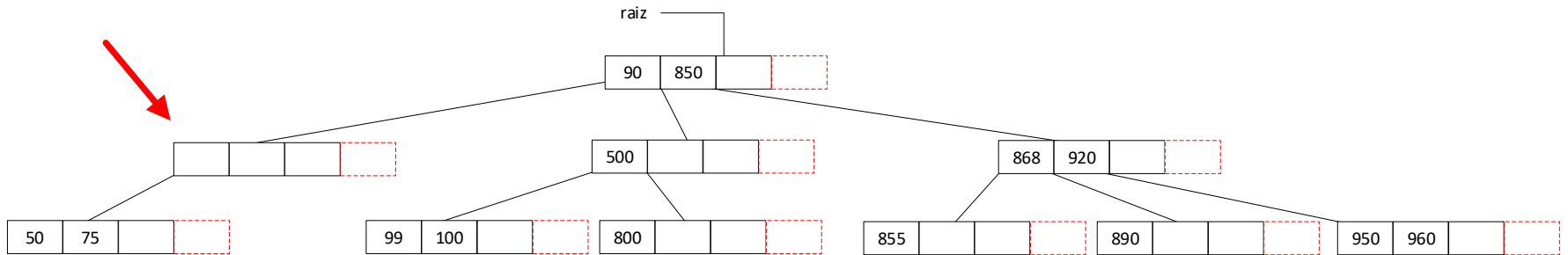
El nodo donde esta el 70 no es hoja, se le busca un reemplazo, el predecesor inOrden, es decir, el 50. Pero esto deja al nodo donde estaba el 50 rompiendo la regla del mínimo de datos

6.1

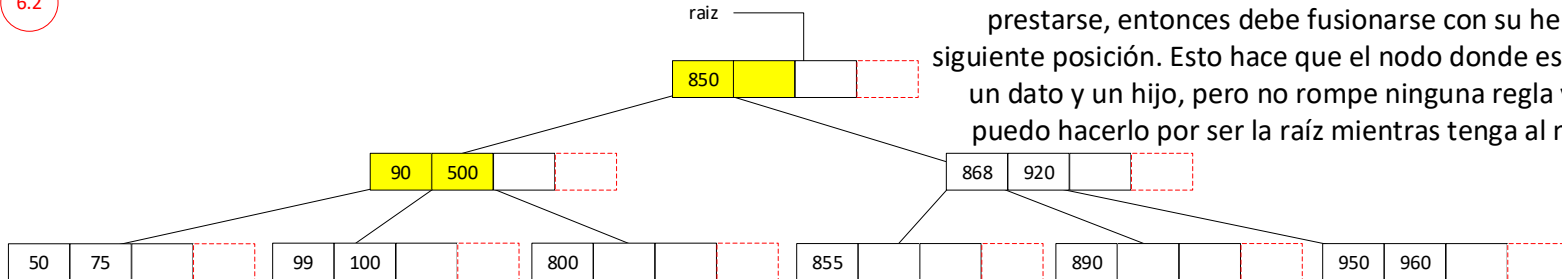


El nodo no tiene de quien prestarse, por lo cual tiene que fusionarse con su hermano. Esto provoca que el padre rompa regla de mínimo de datos e hijos.

ELIMINAR – P.6.



6.2



El nodo padre que rompe las reglas no tiene un hermano de quien prestarse, entonces debe fusionarse con su hermano de la siguiente posición. Esto hace que el nodo donde esta el 850, pierda un dato y un hijo, pero no rompe ninguna regla y si lo hiciera, puedo hacerlo por ser la raíz mientras tenga al menos 2 hijos