*Lo presentado en estas secciones está tomado del material del curso "ISIS1206 – Estructuras de Datos" de la Universidad de Los Andes*

*https://cupintranet.virtual.uniandes.edu.co/sitio/images/cursosCupi2/datos/presentaciones/n16_arbolesnarios.pdf*

## 3.4. ÁRBOLES 1-2-3.

Es un Árbol Triario **ORDENADO**, y en cada nodo tiene 1 ó 2 elementos.

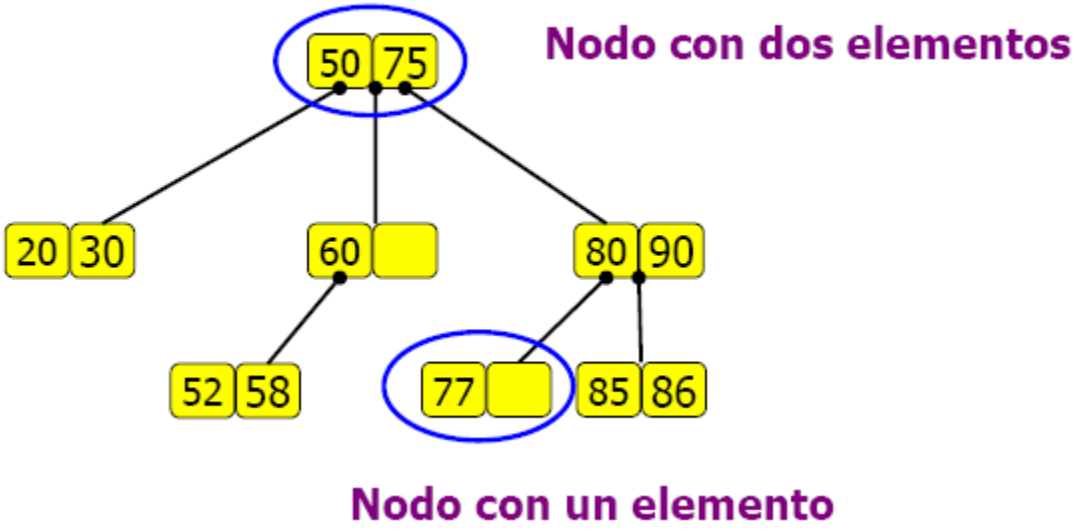



Figura 8.

Cada nodo tiene 1, 2 ó 3subárboles asociados:

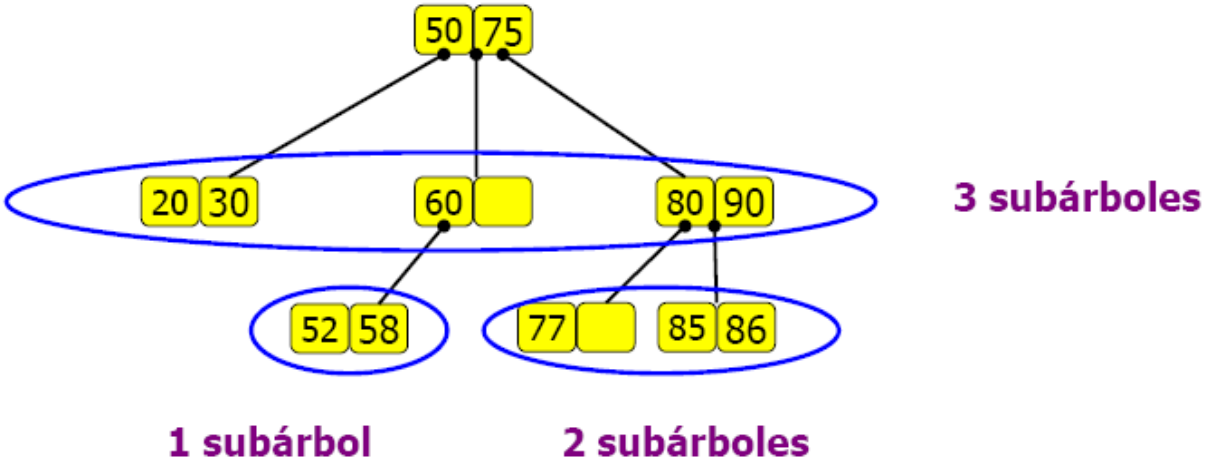



Figura 9.

No hay elementos repetidos.

El elemento de la izquierda de cada nodo (raíz izquierda)

**ES MENOR**

que el elemento de su derecha (raíz derecha)

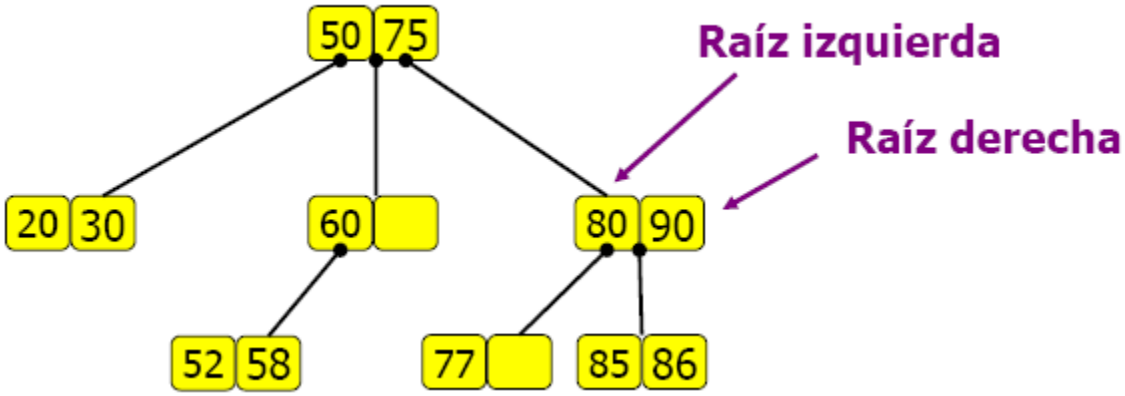



Figura 10.

El primer subárbol es un árbol 1-2-3 que contiene elementos menores que la raíz izquierda:

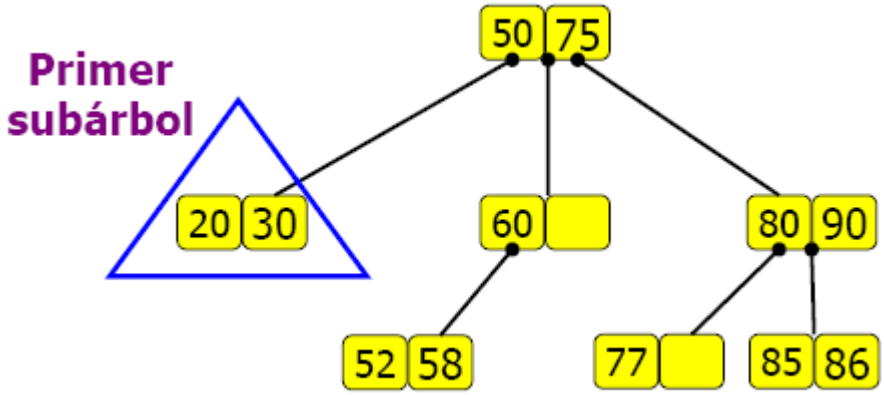



Figura 11.

El segundo subárbol es un árbol 1-2-3 que contiene elementos mayores que la raíz izquierda pero menores que la raíz derecha:

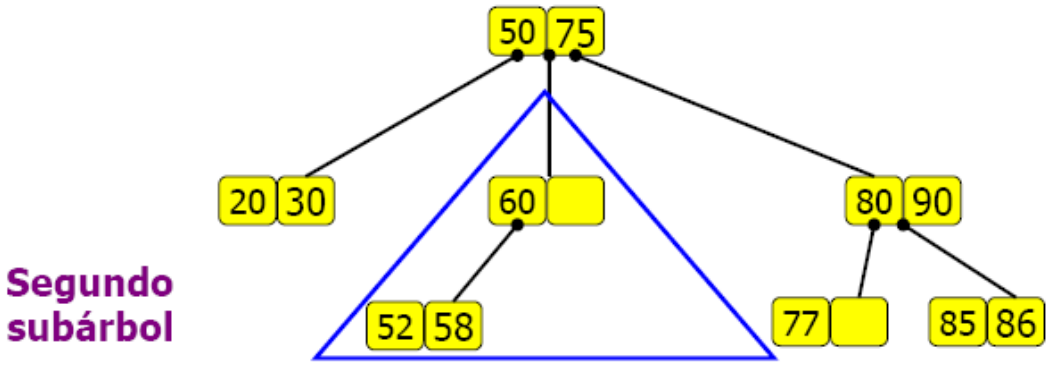



Figura 12.

El tercer subárbol es un árbol 1-2-3 que contiene los elementos mayores que la raíz derecha:

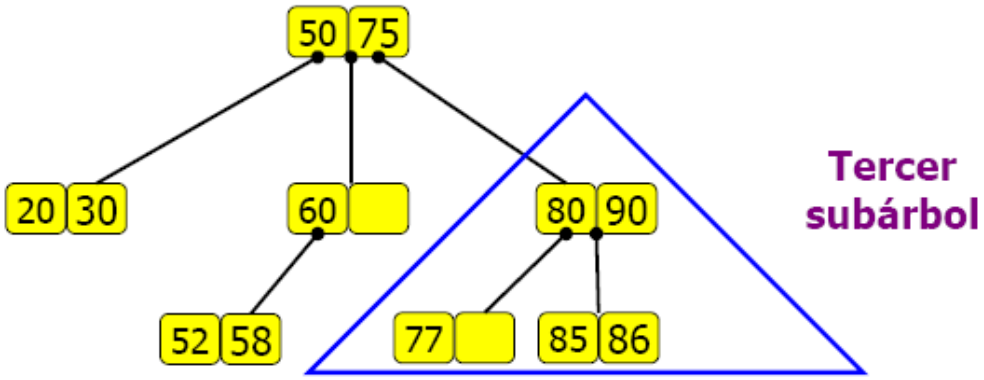



Figura 13.

Si la raíz derecha está vacía, su tercer subárbol debe ser vacío (el segundo puede o no ser vacío):

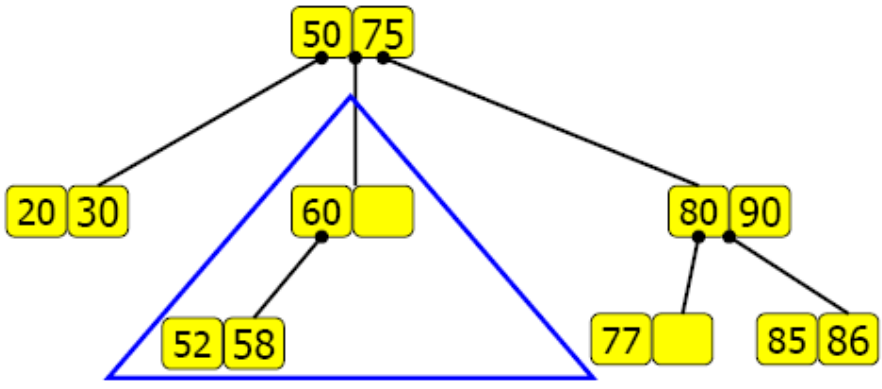



Figura 14.

## 3.5. ÁRBOLES 2-3.

Es un Árbol Triario **ORDENADO Y BALANCEADO.** Al estar balanceado se optimiza el tiempo de acceso en una estructura de datos en memoria secundaria. Acceso a la información en ***O(log$_3$ (N)).***
Los algoritmos de actualización se consideran de baja complejidad.

Es un árbol 1-2-3 que además cuenta con las siguientes características:

- Todas las hojas se encuentran en el mismo nivel.
- Todos los nodos internos tienen por lo menos 2 subárboles asociados no vacíos, aunque la raíz derecha este vacía.

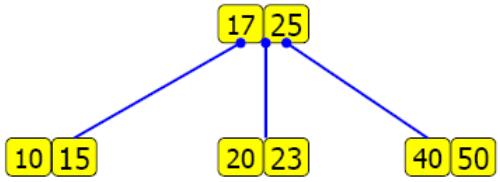



Figura 15.

- Cada nodo pueden tener hasta 2 elementos.
- Un nodo interno puede tener 2 ó 3 hijos, dependiendo de cuántos elementos posea el nodo:
  - Si hay 1 elemento en el nodo, debe tener 2 hijos.
  - Si hay 2 elementos en el nodo, debe tener 3 hijos.

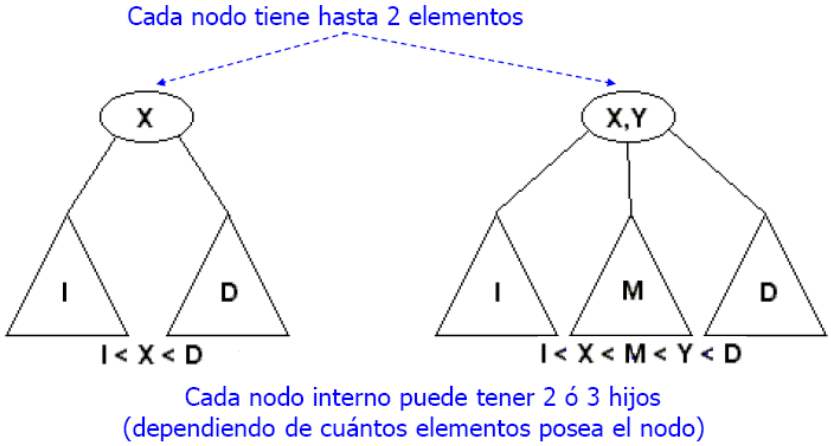



Figura 16.

Formalismo abstracto:

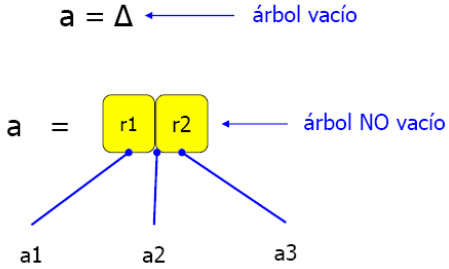



Figura 17.

*OPERACIÓN DE INSERCIÓN EN UN ÁRBOL 2-3:*

- El crecimiento **NO** se hace a nivel de las hojas
  - Aunque la inserción se sigue haciendo en las hojas
- El crecimiento se hace a nivel de la raíz
  - Todas las hojas se deben mantener siempre en el mismo nivel

PASO 1: Localizar **LA HOJA** en la cual se debe agregar el elemento

PASO 2: Insertar. Existen los siguientes casos:

***Caso 1:*** Existe espacio en el nodo, entonces, la estructura del árbol NO se altera.

| SITUACIÓN INICIAL | SITUACIÓN FINAL |
|---|---|
| ***r1*** \| , r1<elem | ***r1*** \| ***elem*** |
| ***r1*** \| , r1>elem | ***elem*** \| ***r1*** |

Figura 18.

***Caso 2:*** El nodo está lleno. Se debe modificar la estructura del árbol:

- El nodo se parte en dos nodos del mismo nivel
- Los tres elementos (dos elementos del nodo y el nuevo elemento) se reparten de la siguiente manera:

| SITUACIÓN INICIAL | SITUACIÓN INTERMEDIA | |
|---|---|---|
| ***r1*** \| ***r2*** , elem>r2 | ***r1*** | ***elem*** , sube **r2** |
| ***r1*** \| ***r2*** , elem<r1 | ***elem*** | ***r2*** , sube **r1** |
| ***r1*** \| ***r2*** , r1<elem<r2 | ***r1*** | ***r2*** , sube **elem** |

*El elemento que no fue incluido en los dos nodos nuevos se sube en la estructura y se inserta en su padre.*

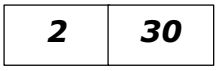
Figura 19.

***Y se repite el proceso hacia arriba:*** Al partir en dos el nodo se está generando un nuevo subárbol que puede generar que los ancestros a su vez tengan que partirse para incluir el elemento requerido.

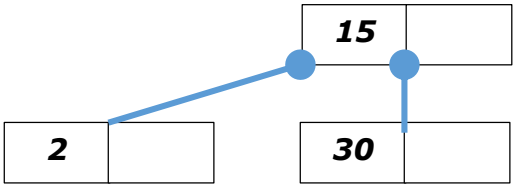
***EJEMPLO PASO A PASO:***

***Insertar (30):*** *Se crea una hoja y se coloca el elemento como* ***raíz izquierda****.*

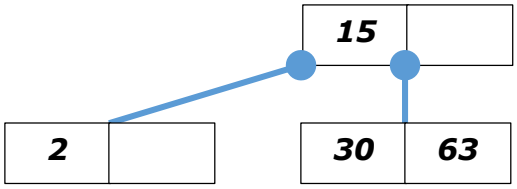

***Insertar (2):*** *Corresponde al caso 1. Se mueve a la derecha la raíz izquierda para dar cabida al nuevo elemento.*




***Insertar (15):*** *Corresponde al caso 2. Encuentra una hoja llena. La parte en dos nodos e inserta en el padre el elemento que se encuentre en la mitad de los tres (2<15<30). Como el padre es vacío, se crea un nuevo nivel, se ubica el elemento como la raíz izquierda del nodo, y se le asocian los dos nodos que se acaban de partir.*

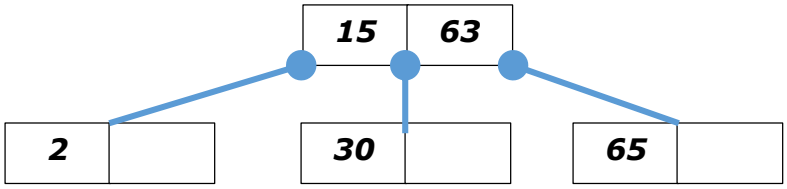



***Insertar (63):*** *Corresponde al caso 1.*

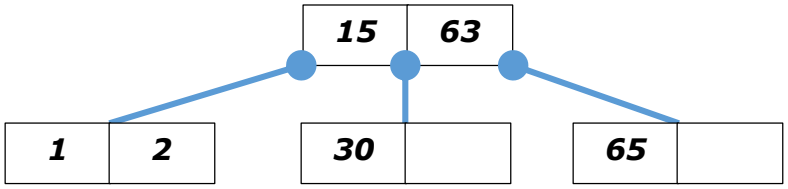



***Insertar (65):*** *Corresponde al caso 2.Se parte la hoja y sube al padre el elemento de la mitad (63).*

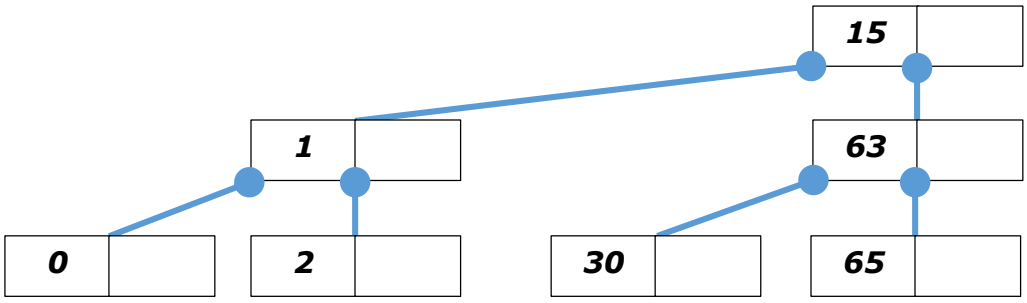
*Al insertar dicho valor en el padre se trata como el caso 1, porque en el nodo hay espacio.*



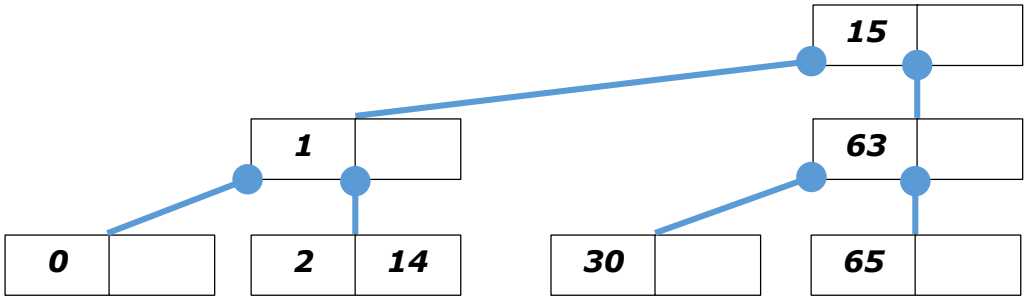

***Insertar (1):*** *Corresponde al caso 1.*



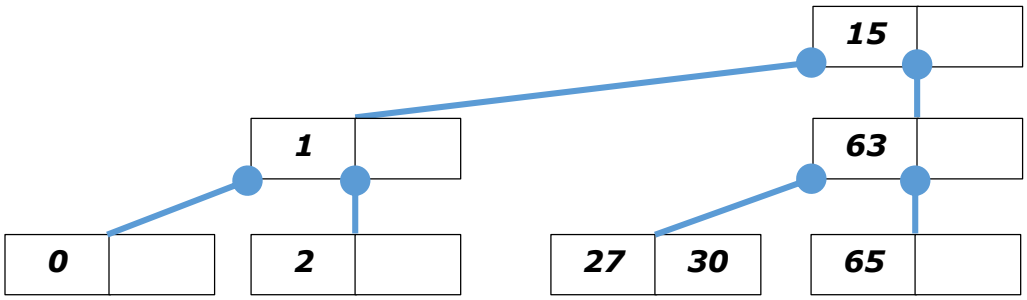
***Insertar (0):*** *Corresponde al caso 2.Se parte la hoja [1,2], se ubican allí los elementos 0 y 2, y sube el valor 1 como su padre. Como el nodo del padre [15,63] está lleno, también se debe partir, dejando en ese nivel los elementos 1 y 63, y subiendo el 15.*



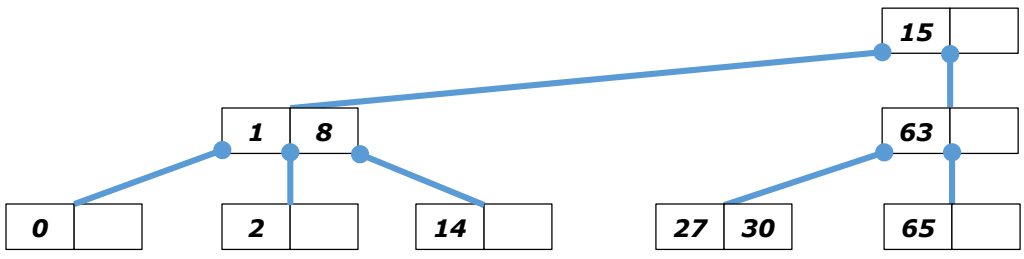

***Insertar (14):*** *Corresponde al caso 1.*



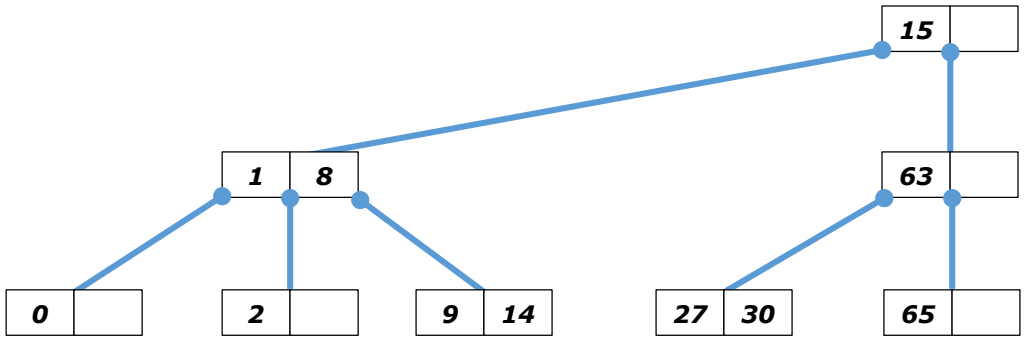

***Insertar (27):*** *Corresponde al caso 1.*



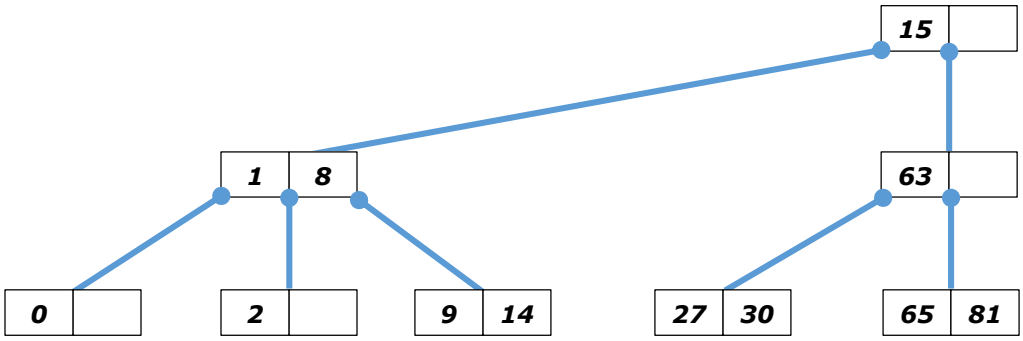

***Insertar (8):*** *Corresponde al caso 2. Se parte el nodo [2,14] y sube el 8. Allí se encuentra espacio y se ubica como raíz derecha.*



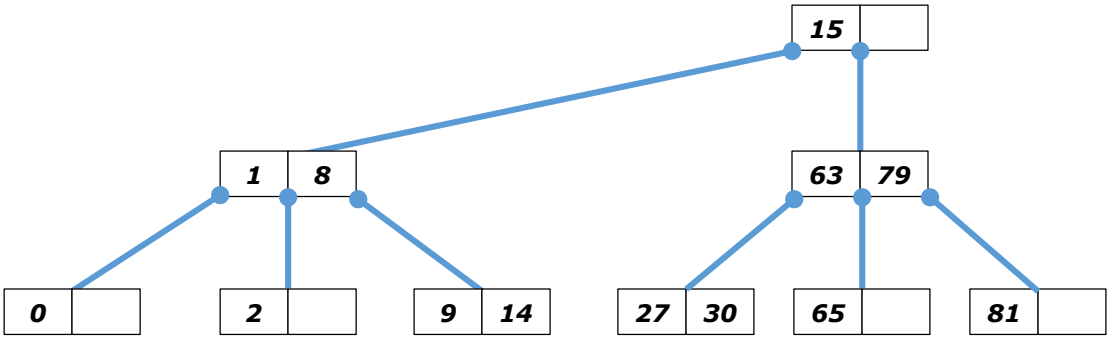
***Insertar (9):*** *Corresponde al caso 1.*



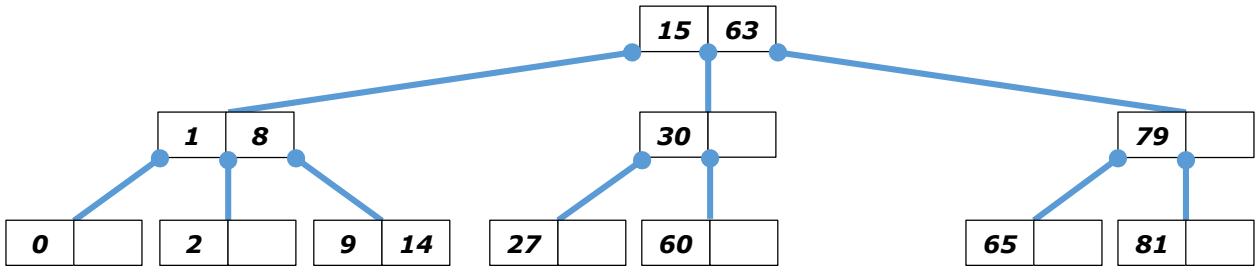

***Insertar (81):*** *Corresponde al caso 1.*




***Insertar (79):*** *Corresponde al caso 2. Se parte el nodo [65,81] y sube el elemento 79.*

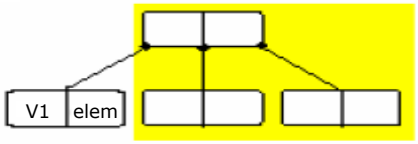
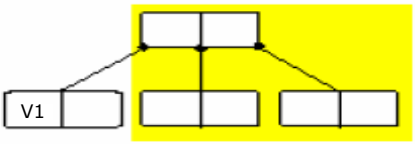
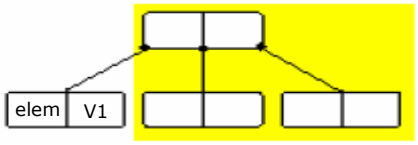
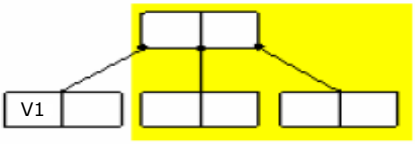
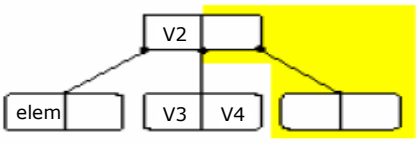
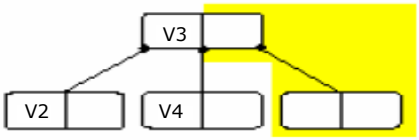
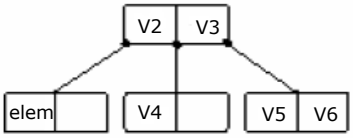
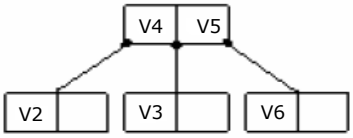
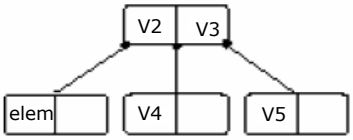
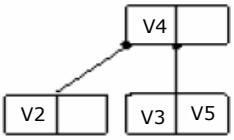
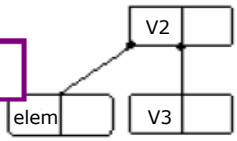
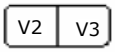
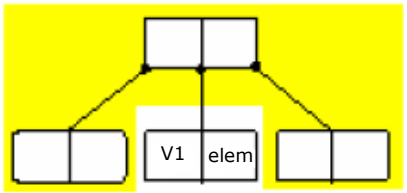
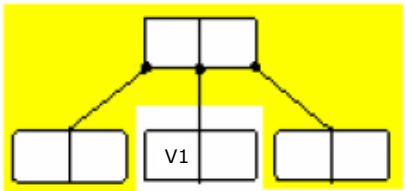
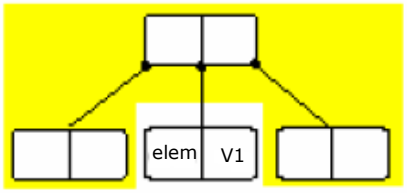
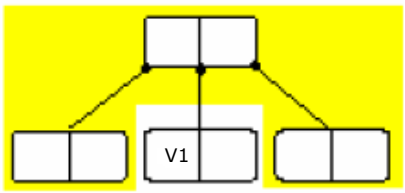



***Insertar (60):*** *Corresponde al caso 2. Se parte el nodo [27,30], se incluye el 60 y sube el elemento 30.*
*Como su padre está lleno, se parte en los nodos [30] y [79], y sube el elemento 63.*
*Este elemento se sitúa en la raíz derecha del árbol, donde hay espacio libre.*

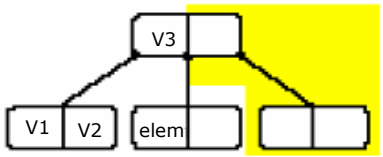
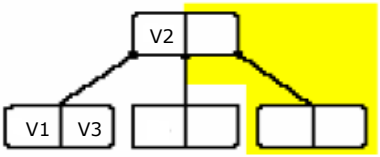
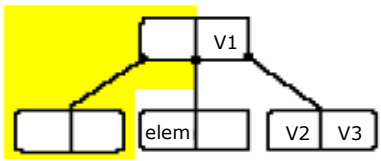
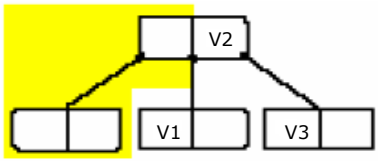
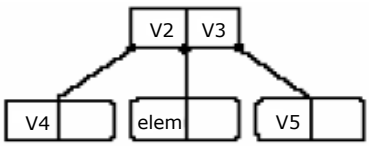
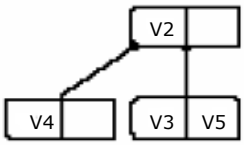
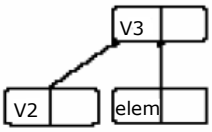
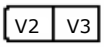
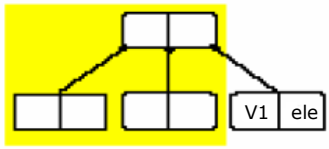
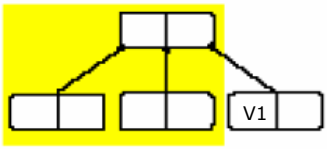
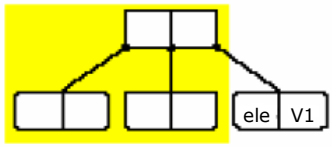
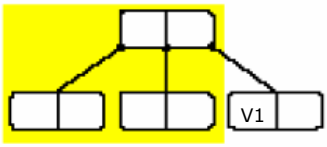
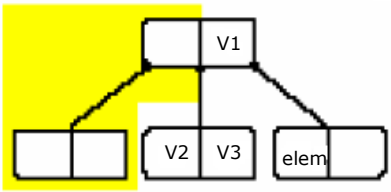
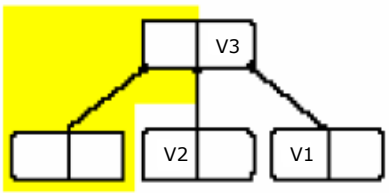
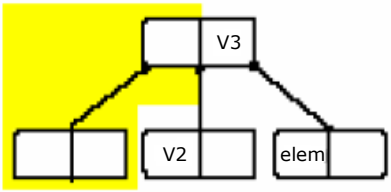
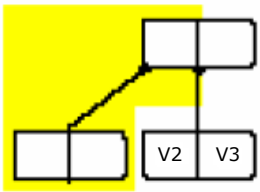


*OPERACIÓN DE ELIMINACIÓN EN UN ÁRBOL 2-3:*

**Caso 1:** El elemento está en una HOJA.

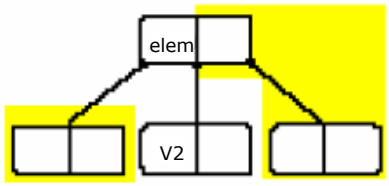
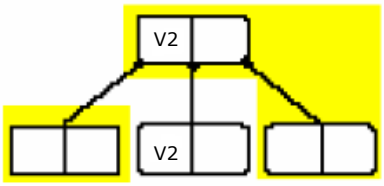
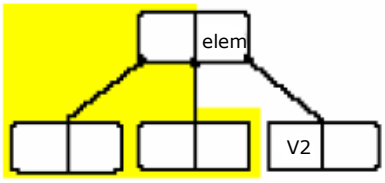
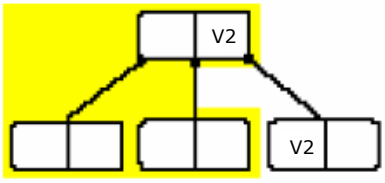
| CASO | Situación inicial | Solución |
|---|---|---|
| A |  |  |
| B |  |  |
| C |  |  |
| D |  |  |
| E |  |  |
| F<br>Se pierde un nivel |  |  |
| G |  |  |
| H |  |  |

**Caso 1:** El elemento está en una HOJA (continuación)

| CASO | Situación inicial | Solución |
|---|---|---|
| I |  |  |
| J |  |  |
| K |  |  |
| L<br>Se pierde un nivel |  |  |
| M |  |  |
| N |  |  |
| O |  |  |
| P |  |  |

**Caso 2:** El elemento no está en una hoja.

Se busca un valor que se encuentre en una hoja y que pueda reemplazar el valor. (Como está ordenado, el candidato sería el menor del subárbol derecho).

| CASO | Situación inicial | Situación intermedia: Eliminar v2 de una hoja |
|---|---|---|
| Q |   |   |
| R |   |   |

EJEMPLO DE ELIMINACIÓN: Retomando el árbol del ejemplo de inserción paso a paso, y queremos eliminar el 63.

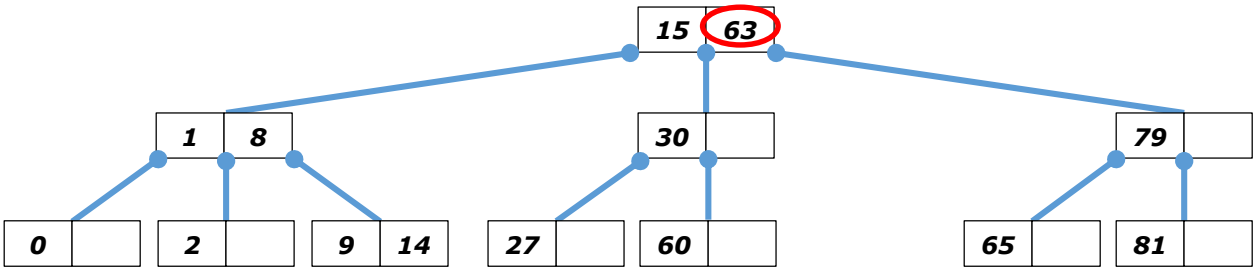
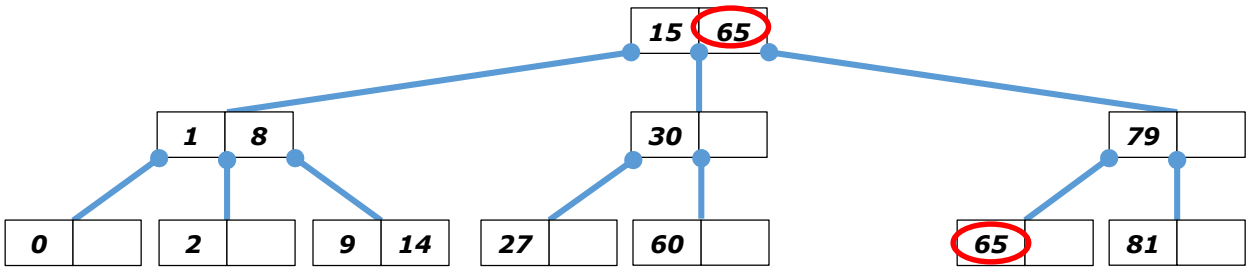
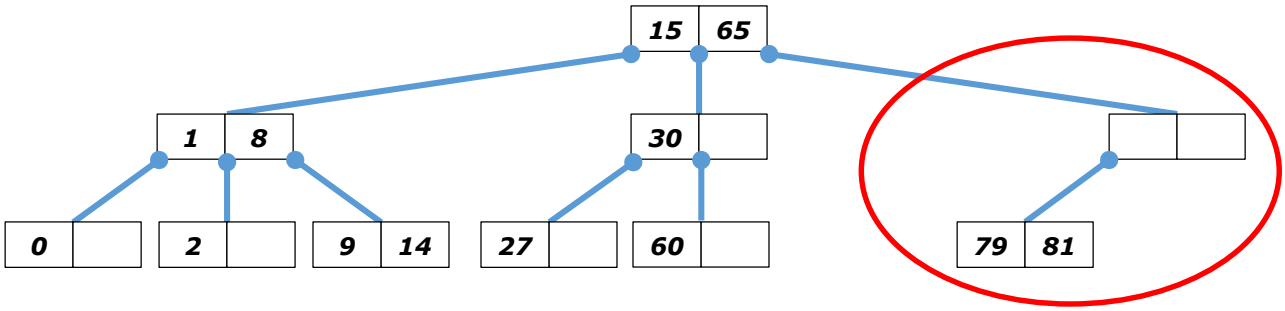



Figura 20.

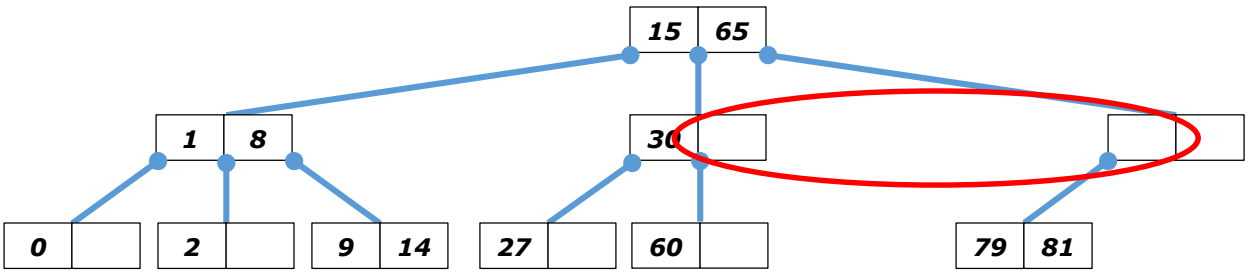
***(1)*** *Se busca un valor en una hoja que pueda ocupar el espacio que va a liberar el 63.*
*Se escoge para esto el elemento 65 (caso 2-R), y se replantea el problema para eliminar ese valor.*



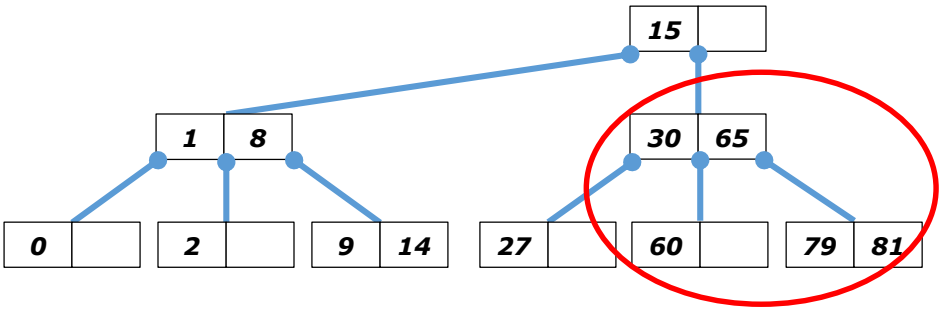
***(2)*** *El tercer subárbol entra en el caso 1-F, en el cual se pierde un nivel. Se llega al siguiente caso intermedio, porque el árbol debe decrecer por la raíz.*




***(3)*** *El árbol completo llega al caso 1-P, en el cual el segundo subárbol NO tiene raíz derecha para prestársela al tercer subárbol. Se debe aplicar la solución planteada para ese caso, pero teniendo en cuenta que el elemento no es una hoja, y que también deben moverse los subárboles asociados.*




***(4)*** *El proceso termina porque el árbol ya es correcto estructuralmente.*




---------- FIN DE ESTA SECCIÓN