



UNIVERSIDAD DE EL SALVADOR  
FACULTAD DE INGENIERIA Y ARQUITECTURA  
ESCUELA DE INGENIERIA DE SISTEMAS INFORMATICOS  
ESTRUCTURAS DE DATOS

## ARBOLES BALANCEADOS

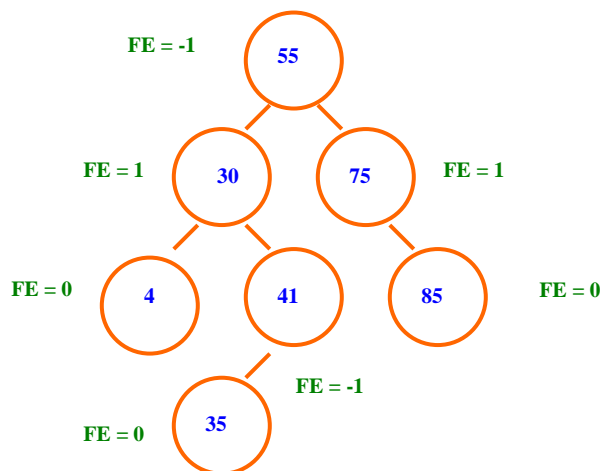
Un árbol balanceado es un árbol binario en el cual las alturas de los dos subárboles para cada nodo nunca difieren en más de una unidad. También se les llama árboles **AVL** en honor a sus inventores, dos matemáticos rusos, G. M. **A**delson-**V**elskii y E.M. **L**andis.

El Factor de Equilibrio (FE) o de Balance (FB) de un nodo se define como la altura del SAD menos la altura del SAI correspondiente. El Factor de Equilibrio de cada nodo en un árbol balanceado será  $-1$ ,  $1$  ó  $0$ . Si FE llegara a tomar los valores de  $-2$  ó  $2$ , entonces debería reestructurarse el árbol.

FE expresado en fórmula, sería:

$$FE = h_{SAD} - h_{SAI}$$

La figura muestra un árbol binario balanceado (AVL), con los factores de equilibrio calculados para cada nodo:



### 1. Inserción de Nodos en Arboles AVL

Al insertar un elemento en un árbol balanceado deben distinguirse los siguientes casos:

CASO 1. El SAI y el SAD del árbol balanceado tienen la misma altura ( $h_{SAD} = h_{SAI}$ ):

- a) Si se inserta un elemento en SAI entonces  $h_{SAD}$  será menor que  $h_{SAI}$
- b) Si se inserta un elemento en SAD entonces  $h_{SAD}$  será mayor que  $h_{SAI}$

Ya sea para a) o para b), no se viola el criterio de equilibrio o balance del árbol.

CASO 2. El SAI y el SAD del árbol balanceado tienen altura diferente ( $h_{SAD} \neq h_{SAI}$ ):

CASO 2.1. Si  $h_{SAD} > h_{SAI}$

- a) Si se inserta un elemento en SAI entonces  $h_{SAD}$  será igual a  $h_{SAI}$   
Las ramas tienen la misma altura por lo que se mejora el equilibrio
- b) Si se inserta un elemento en SAD entonces el árbol debe ser reestructurado  
Las ramas están desequilibradas por lo que se requiere reestructuración

CASO 2.2. Si  $h_{SAD} < h_{SAI}$

- a) Si se inserta un elemento en SAI entonces el árbol debe ser reestructurado  
Las ramas están desequilibradas por lo que se requiere reestructuración
- b) Si se inserta un elemento en SAD entonces  $h_{SAD}$  será igual a  $h_{SAI}$   
Las ramas tienen la misma altura por lo que se mejora el equilibrio

Para poder determinar si un árbol está balanceado debe calcularse el FE de cada nodo del árbol.

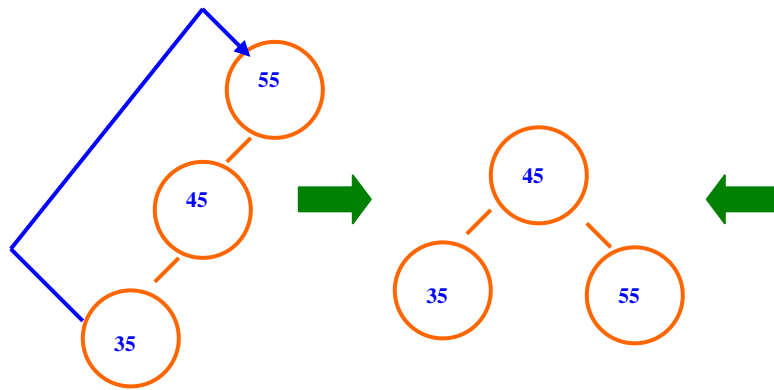
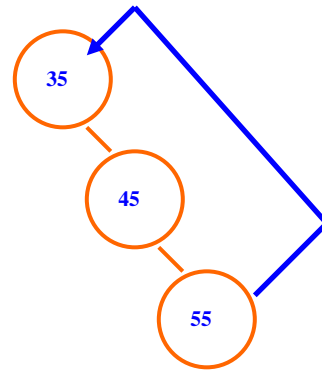
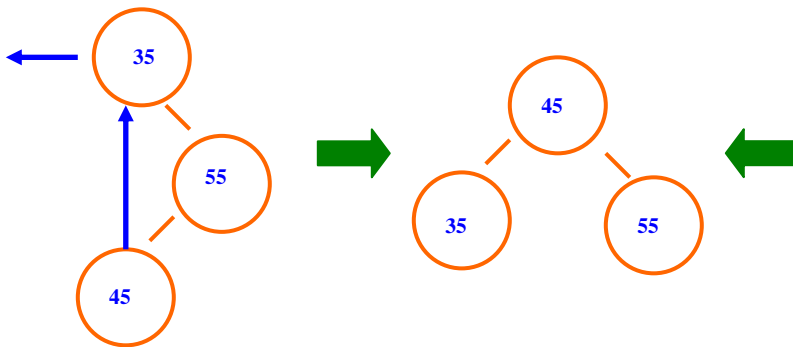
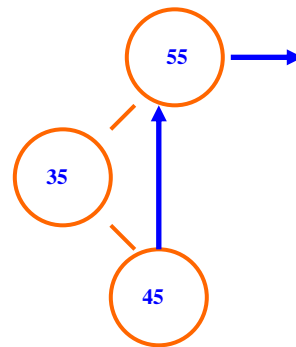
El proceso de inserción en un árbol balanceado consta de los siguientes pasos:

1. Primero debe seguirse el camino de búsqueda del árbol, hasta localizar el lugar donde hay que insertar el elemento.
2. Se calcula su FE, que será cero (pues el elemento recién insertado posee SAI y SAD vacíos). Luego, se regresa por el camino de búsqueda calculando el FE de todos los demás nodos que componen el árbol. Si en alguno de los nodos se viola el criterio de equilibrio entonces debe reestructurarse el árbol. El proceso termina al llegar a la raíz del árbol, o cuando se realiza la reestructuración del mismo, en cuyo caso no es necesario determinar el FE de los nodos restantes.

## 2. Reestructuración de Arboles AVL

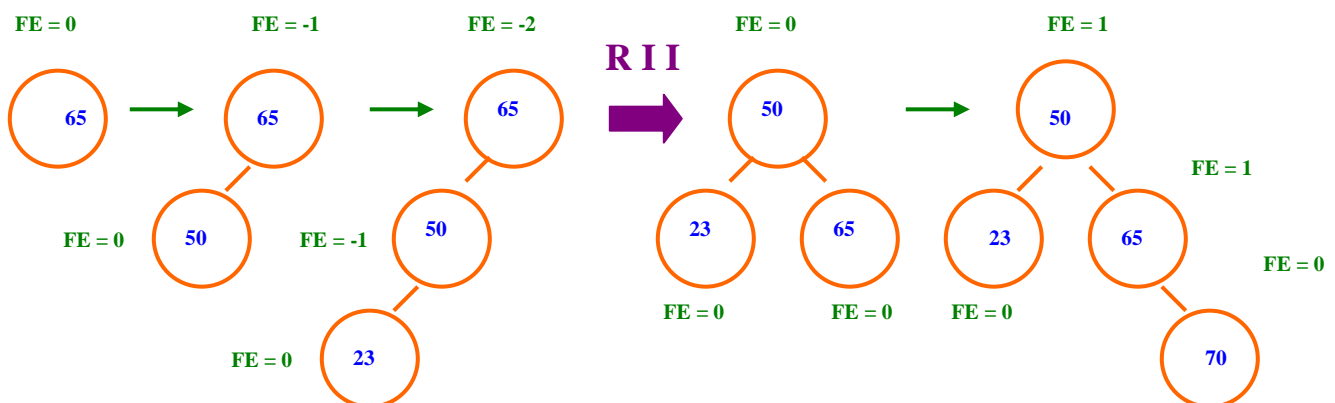
Reestructurar un árbol balanceado significa rotar los nodos del mismo. La rotación puede ser simple o compuesta. En el primer tipo de rotación se involucran dos nodos y en el segundo, se afectan tres. Si la rotación es simple puede realizarse por las ramas derechas (RDD: Rotación Derecha Derecha) o por las ramas izquierdas (RII: Rotación Izquierda Izquierda). Si la rotación es compuesta puede realizarse por las ramas derecha e izquierda (RDI: Rotación Derecha Izquierda) o por las ramas izquierda y derecha (RID: Rotación Izquierda Derecha).

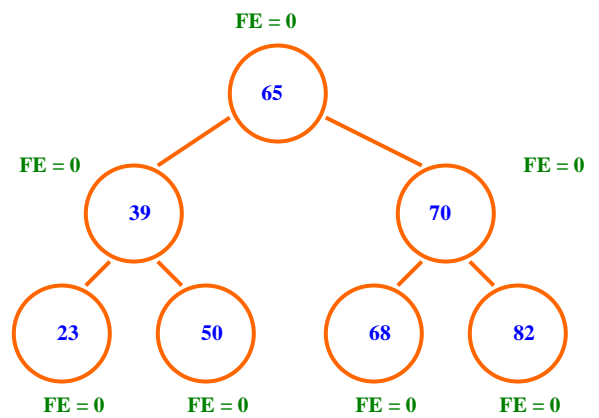
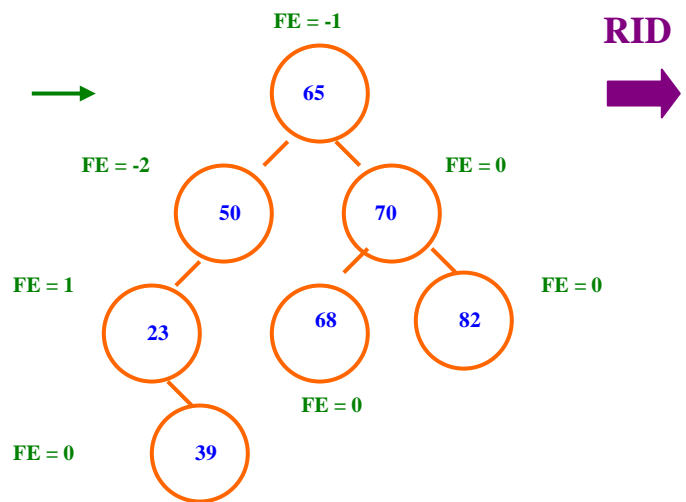
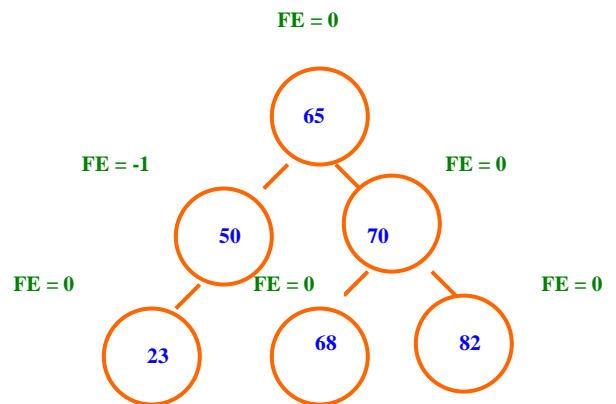
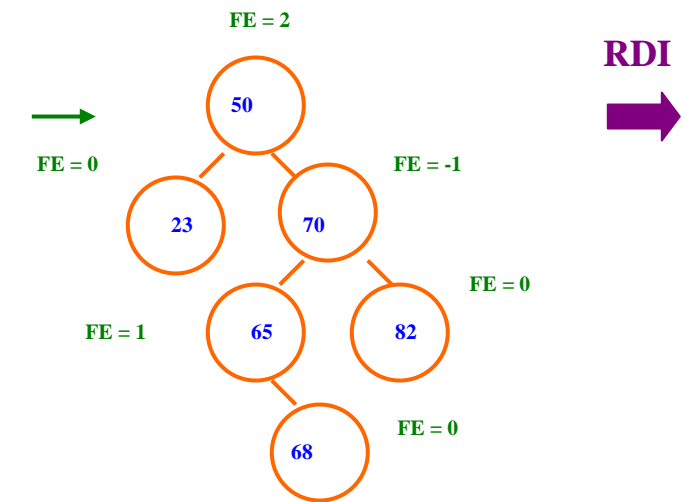
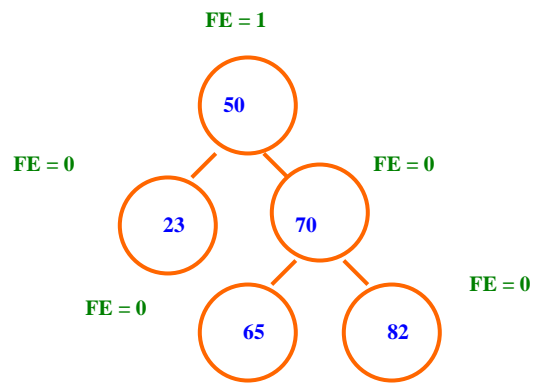
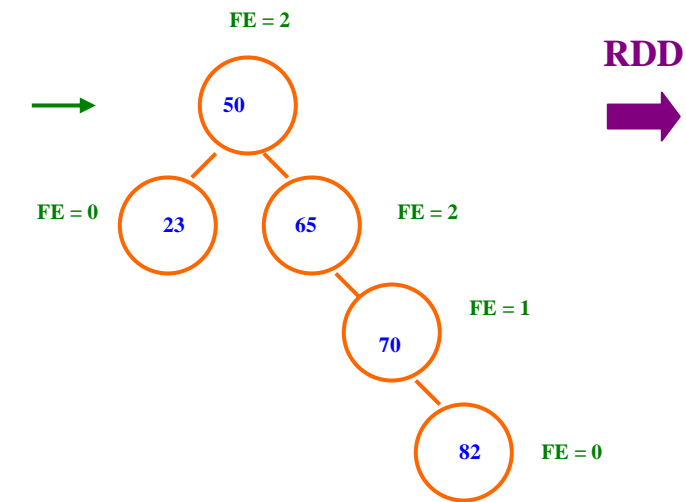
Las siguientes figuras muestran gráficamente los diferentes tipos de rotaciones:

**ROTACION II****ROTACION DD****ROTACION DI****ROTACION ID****3. Ejemplo de Inserción de Nodos en Arboles AVL**

Insertar la siguiente secuencia de claves en un árbol balanceado vacío:

65, 50, 23, 70, 82, 68 y 39





## 4. Eliminación de Nodos en Árboles AVL

La operación de borrado en árboles balanceados consiste en quitar un nodo del árbol sin violar los principios que definen a un árbol balanceado.

En la operación de borrado en árboles balanceados se distinguen los siguientes casos:

CASO 1. Si el elemento a borrar es hoja, simplemente se suprime.

CASO 2. Si el elemento a borrar tiene sólo un hijo, entonces tiene que sustituirse por él.

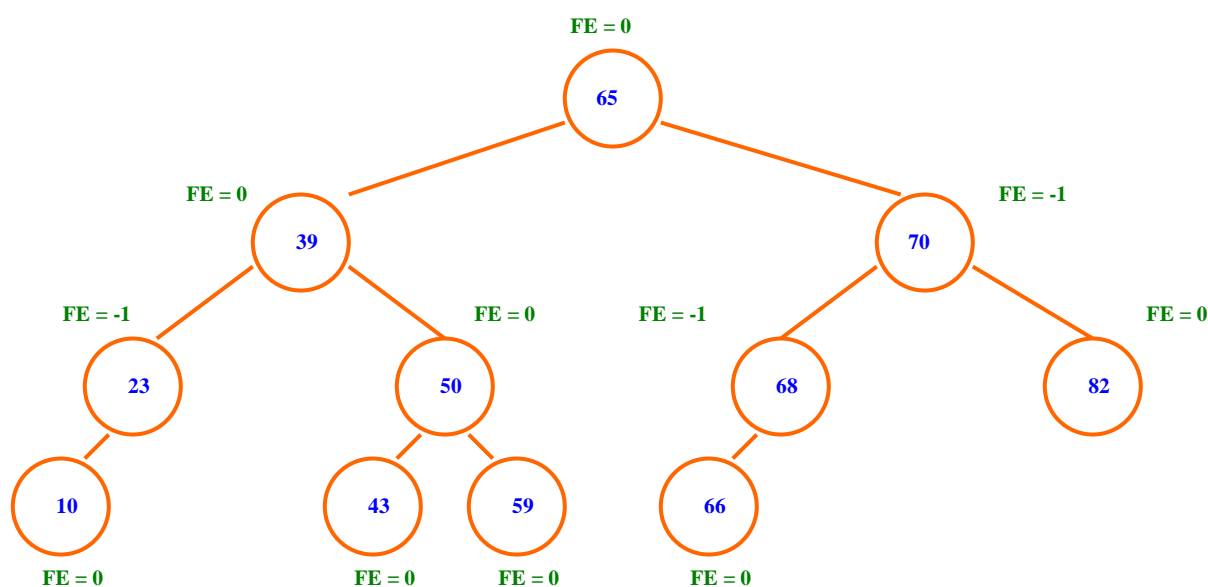
CASO 3. Si el elemento a borrar tiene los dos hijos, entonces tiene que sustituirse por el nodo que se encuentra más a la izquierda en el SAD o por el nodo que se encuentra más a la derecha en el SAI.

Para eliminar un nodo en un árbol balanceado lo primero que debe hacerse es localizar su posición en el árbol. Se elimina siguiendo los criterios establecidos anteriormente y se regresa por el camino de búsqueda calculando el FE de los nodos visitados. Si en alguno de los nodos se viola el criterio de equilibrio, entonces debe reestructurarse el árbol. El proceso termina cuando se llega hasta la raíz del árbol.

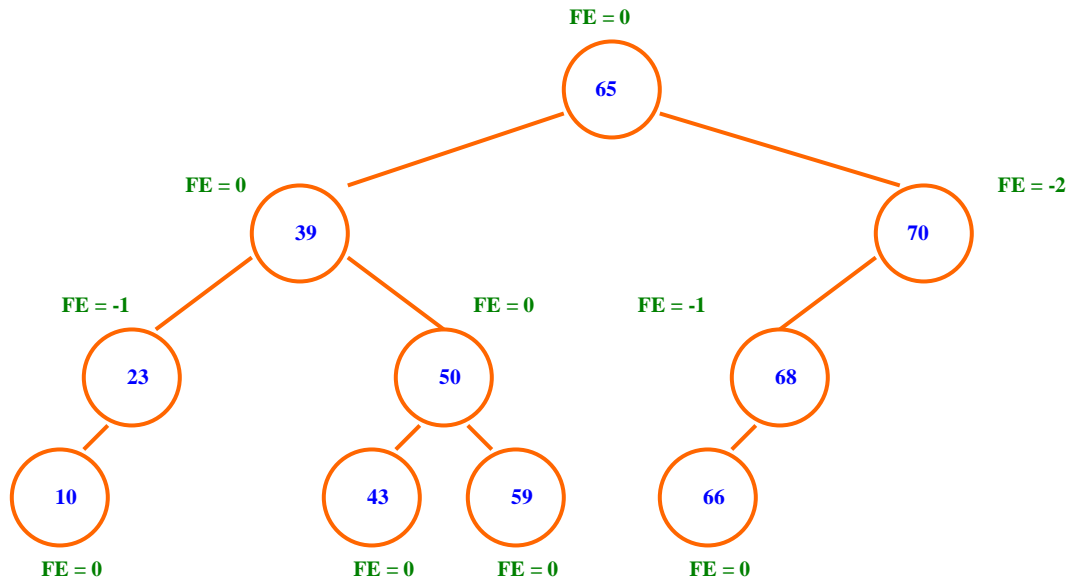
## 5. Ejemplo de Eliminación de Nodos en Árboles AVL

Eliminar las siguientes claves del árbol balanceado de la figura:

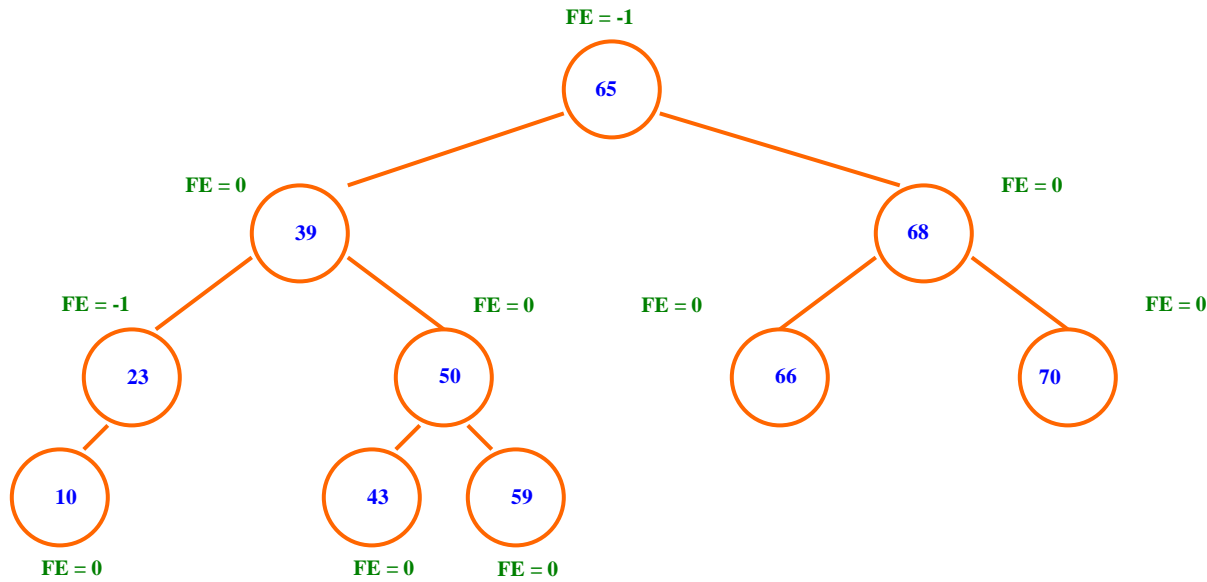
82, 10, 39, 65, 70, 68 y 66



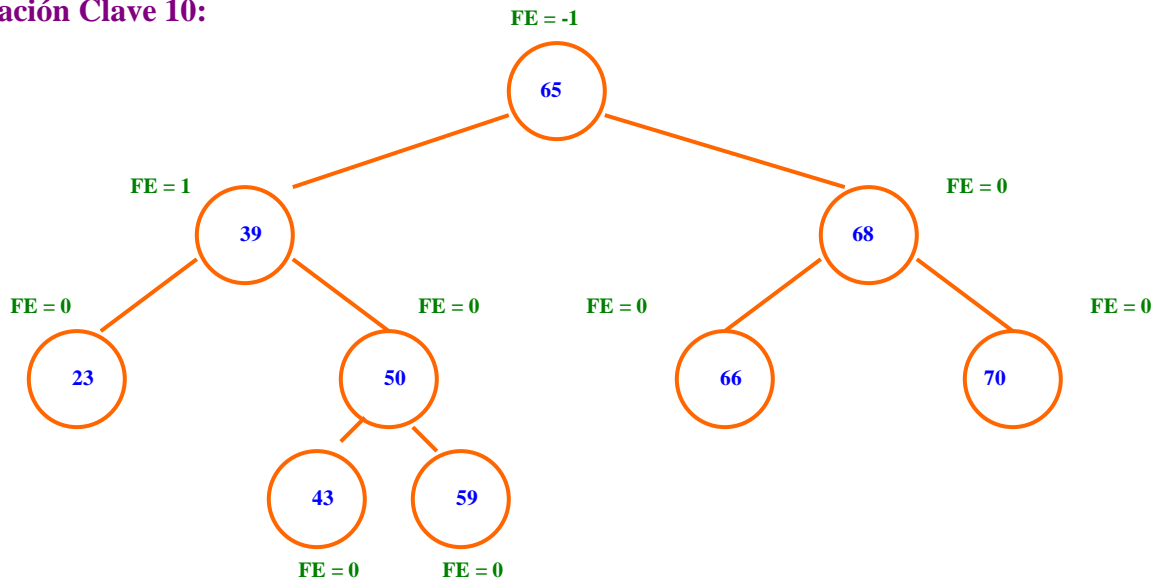
Eliminación Clave 82:



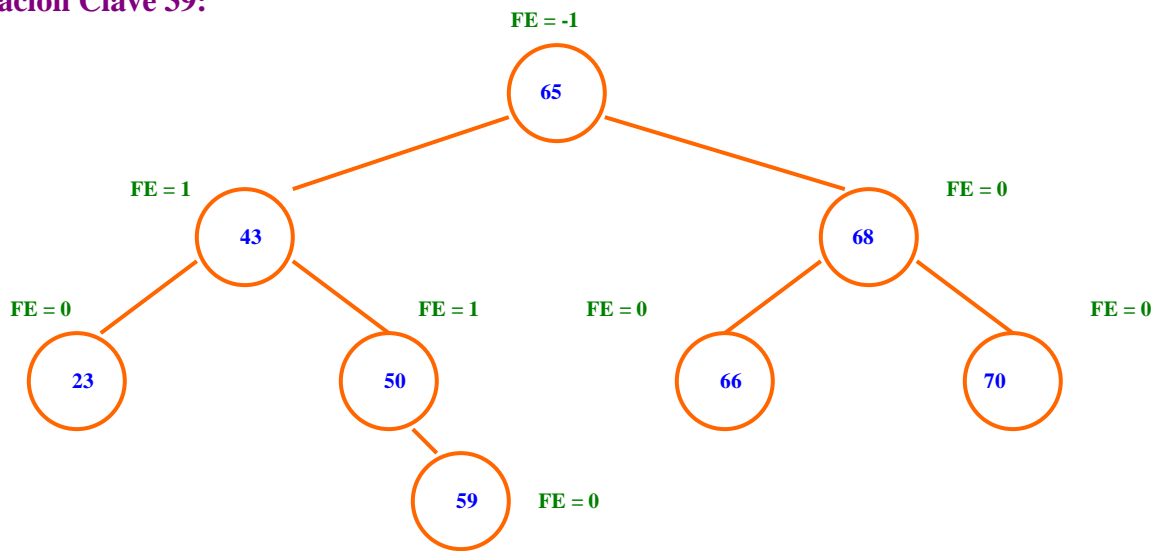
Rotación Izquierda Izquierda (RII):



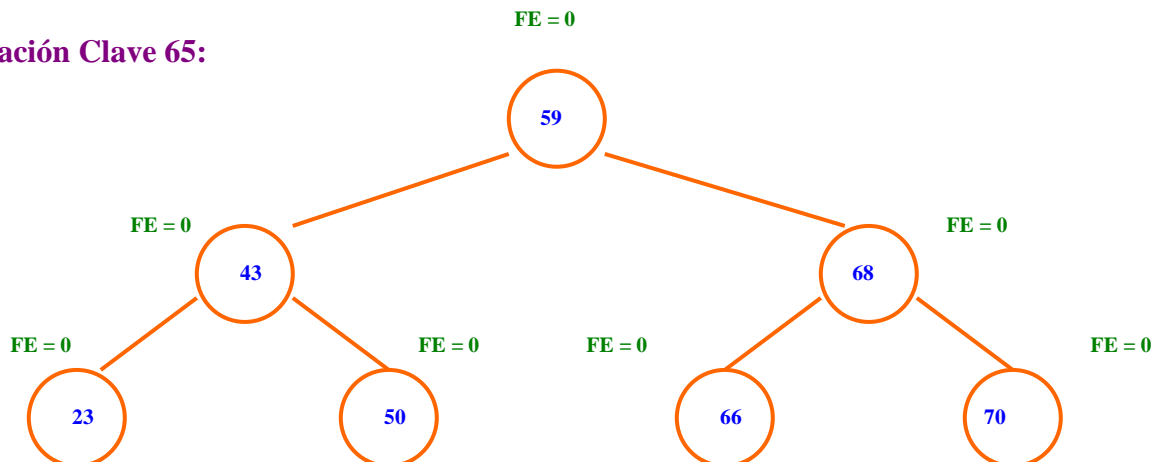
### Eliminación Clave 10:



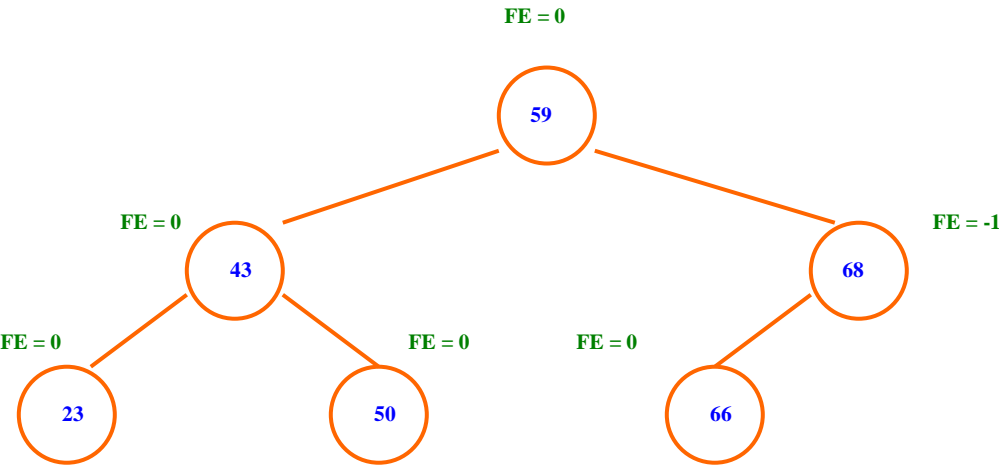
### Eliminación Clave 39:



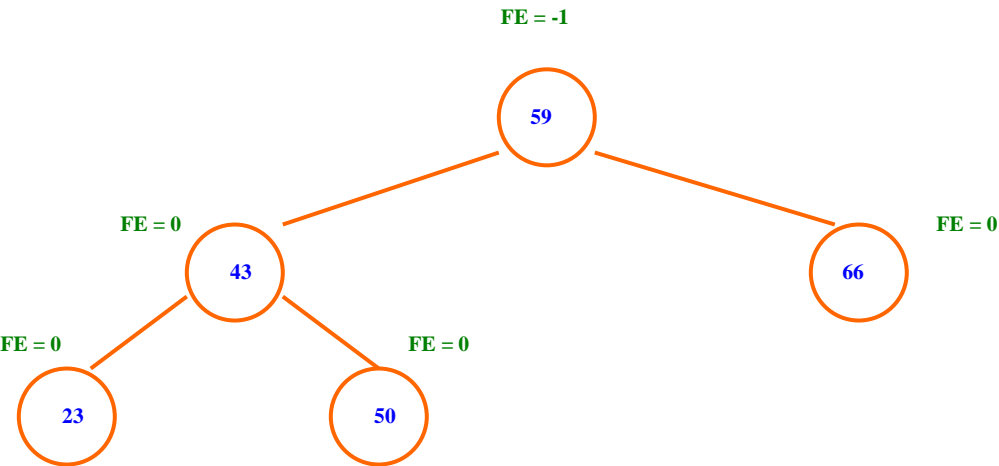
### Eliminación Clave 65:



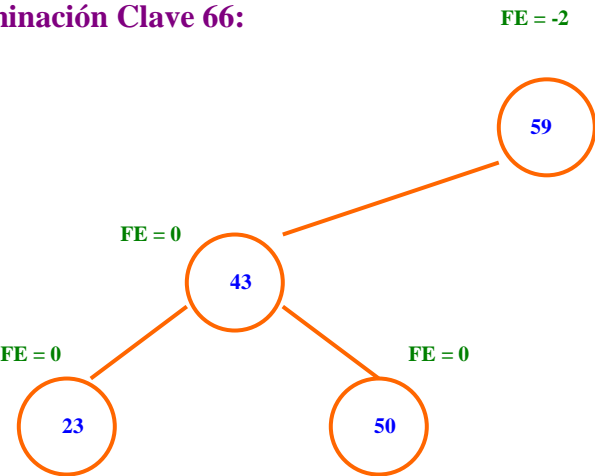
Eliminación Clave 70:



Eliminación Clave 68:

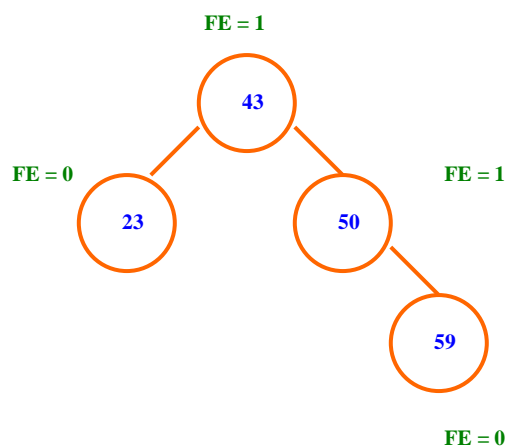


Eliminación Clave 66:

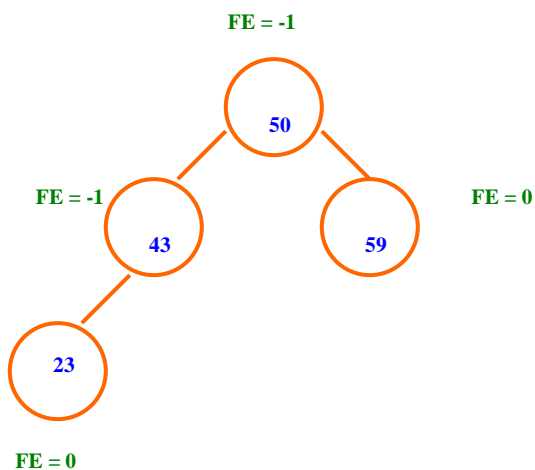




Se puede reestructurar con RII:



También se puede reestructurar con RID:



**Ambas Respuestas son Válidas**

## 6. Ejercicios

6.1. Construir un árbol binario balanceado para almacenar los siguientes datos:

- a) 86, 65, 70, 67, 73, 69, 25, 66, 47, 68, 62, 10, 60 y 93
- b) 22, 43, 56, 65, 87, 93, 99, 120, 130, 135, 140
- c) 31, 12, 20, 03, 24, 23, 21, 02, 19, 65 y 67

6.2. Realizar los recorridos de cada uno de los árboles construídos anteriormente:

- a) Pre-Orden
- b) En-Orden
- c) Post-Orden

6.3. Del segundo árbol creado en 4.1., eliminar la siguiente secuencia de claves:

22, 99, 87, 120, 140, 135, 56