

Escrito por **Adrián Quiroga Linares**.

Con **árboles desbalanceados** se pierda la eficiencia de búsqueda, acercándose a la de una lista, es decir $O(n)$ para la búsqueda. Sin embargo, con árboles equilibrados se consigue una eficiencia de $O(\log(n))$ en la búsqueda, inserción o eliminación.

Aquellos árboles que se mantiene siempre equilibrados debido a constantes reordenaciones se conocen como **árboles AVL**.

En este tipo de árboles, cada nodo cuenta con un campo llamado **factor de equilibrio**, que almacena la diferencia entre la altura del subárbol derecho y la altura del subárbol izquierdo para dicho nodo. Un árbol será equilibrado mientras el valor absoluto de todos los factores de carga sea **menor que 2**.

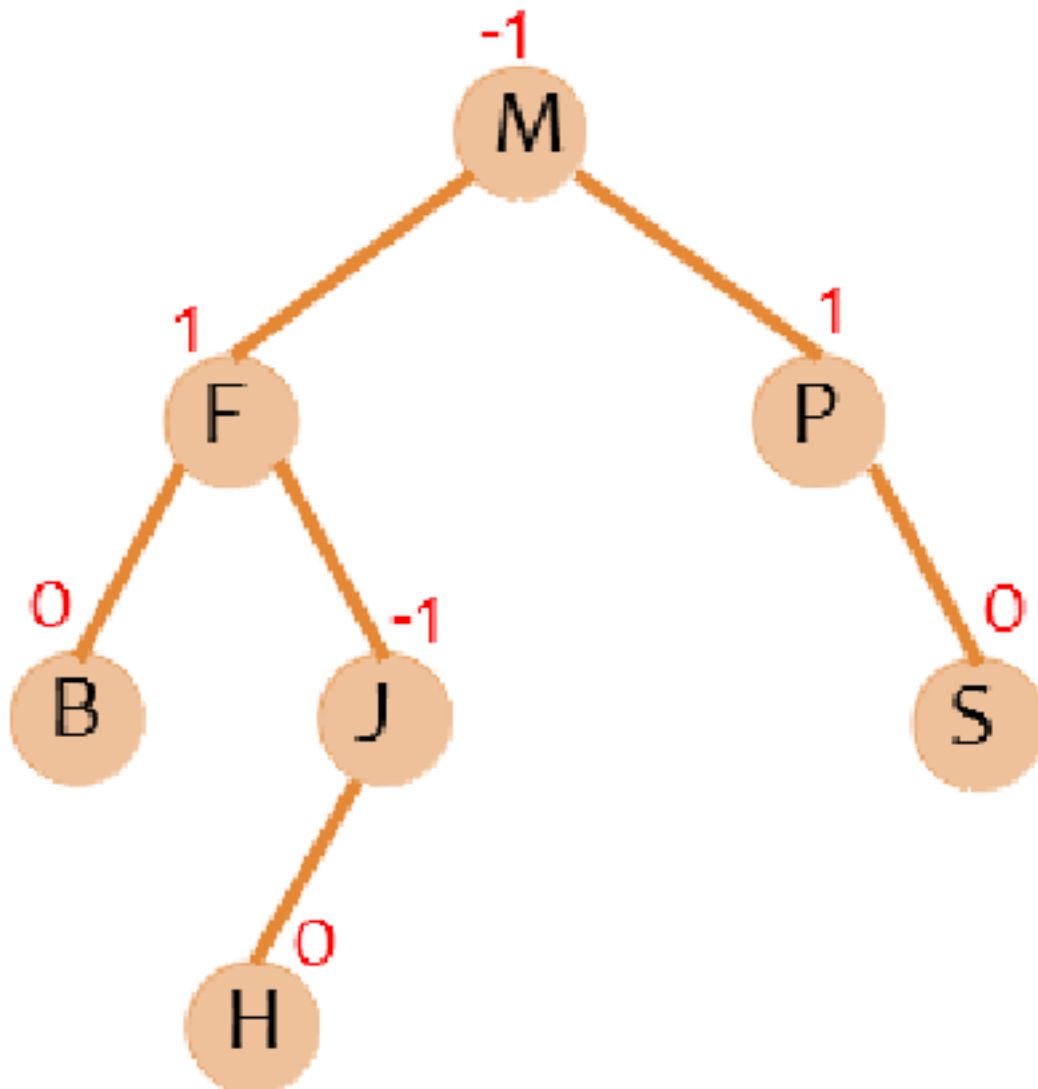


Figure 1: archivos/imagenes/Pasted image 20241002193941.png

Inserción

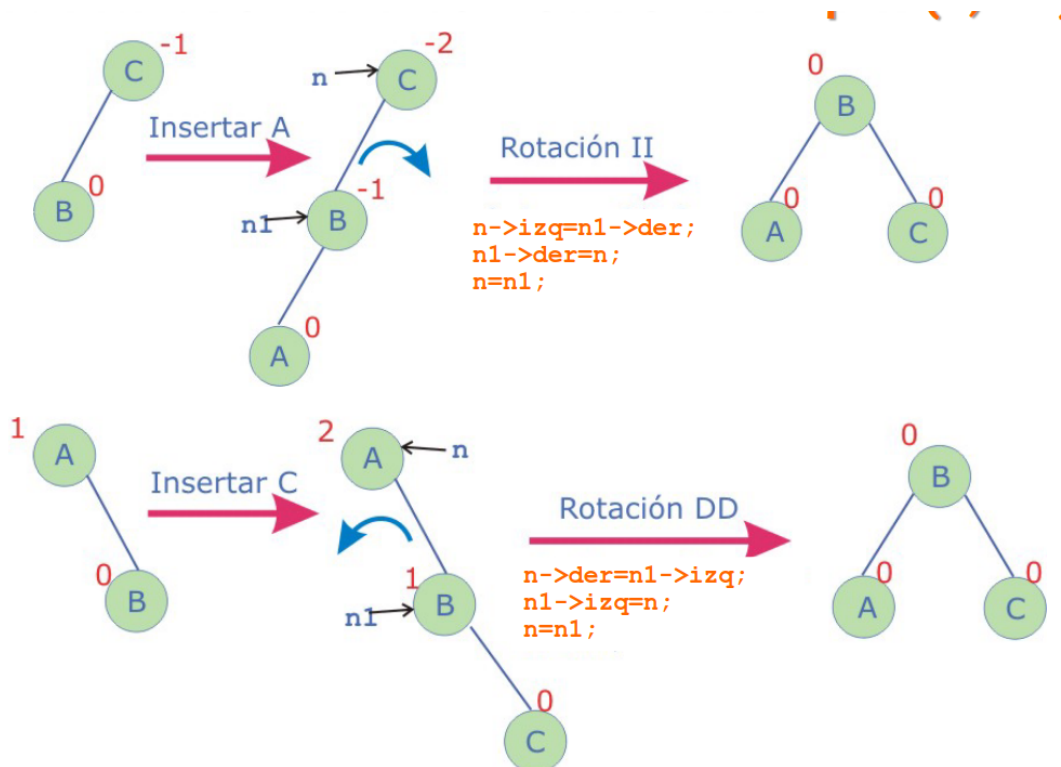
Se pueden dar varios casos en la inserción: - Si las ramas **izquierda y derecha tienen misma altura**, da igual donde se inserte, que al insertar solo se producirá un nuevo **factor de equilibrio de 1 o -1**, con lo que el árbol sigue estando equilibrado. - Si las ramas **izquierda y derecha difieren en su altura en 1 unidad**:

- a) Si el factor de equilibrio era **-1** se inserta por la **derecha**, o si el factor de equilibrio era **1** se inserta por la **izquierda**, el nuevo factor de equilibrio pasa a ser **0**, por lo que se mejora el equilibrio del árbol y no hay ningún problema.
- b) Si el factor de equilibrio era **-1** y se inserta por la **izquierda**, o si el nuevo factor de equilibrio era **1** y se inserta por la **derecha**, el nuevo factor de equilibrio pasa a ser **-2 o 2** respectivamente, desequilibrando el árbol, y siendo necesaria una reestructuración.

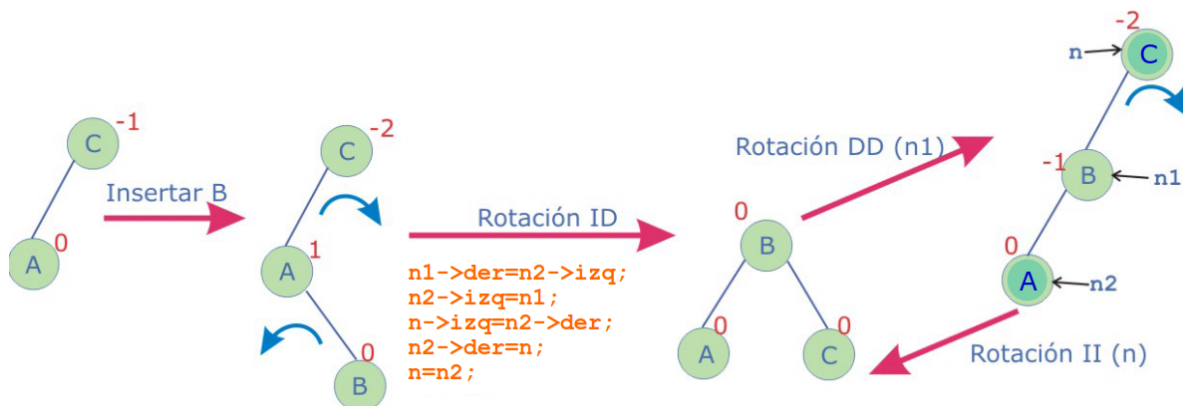
Reestructuración

Tras una inserción, el nodo insertado pasa a ser **un nodo hoja**, por lo que su factor de equilibrio es 0. Posteriormente vamos subiendo por el árbol **recalculando los factores de equilibrio** hasta encontrar uno que cuyo valor absoluto sea 2, a partir del cual habrá que reestructurar el árbol. Así hasta llegar al nodo raíz.

4 tipos de rotaciones, I-> Izquierda D-> Derecha: - **Rotación simple II:** Se produce cuando un nodo tiene **factor -2** y su hijo izquierdo **factor -1**. Se rota hacia la **derecha**, dejando el nodo con factor **-1** como **nuevo padre**, el nodo con factor **-2** como **hijo derecho**, y el nodo que originó el desequilibrio como **hijo izquierdo**. - **Rotación simple DD:** Se produce cuando un nodo tiene **factor 2** y su hijo derecho es **factor -1**. Es lo contrario al caso anterior: Rotación a la izquierda, nodo con **factor 1** como **padre**, nodo con **factor 2** como **hijo izquierdo**, y nodo que originó el desequilibrio como **hijo derecho**.



- **Rotación compuesta DI:** se produce cuando un nodo tiene **factor 2** y su **hijo derecho, factor -1**. Se rota el hijo con factor a la derecha y se sube el nodo que originó el desequilibrio a su posición, y se tiene una **rotación simple DD**. ![[Archivos/imagenes/Pasted image 20241002201001.png]]
- **Rotación compuesta ID:** se produce cuando un nodo tiene **factor -2** y su **hijo izquierdo, factor 1**. Se rota el hijo con factor **1** a la izquierda y se pone el nodo que causó el desequilibrio en su posición, teniendo una **rotación simple II**.



$n \rightarrow fe$	$n1 \rightarrow fe$	Rotación
+2	≥ 0	D D
	< 0	D I
-2	≤ 0	I I
	> 0	I D

Figure 2: archivos/imagenes/Pasted image 20241002201120.png

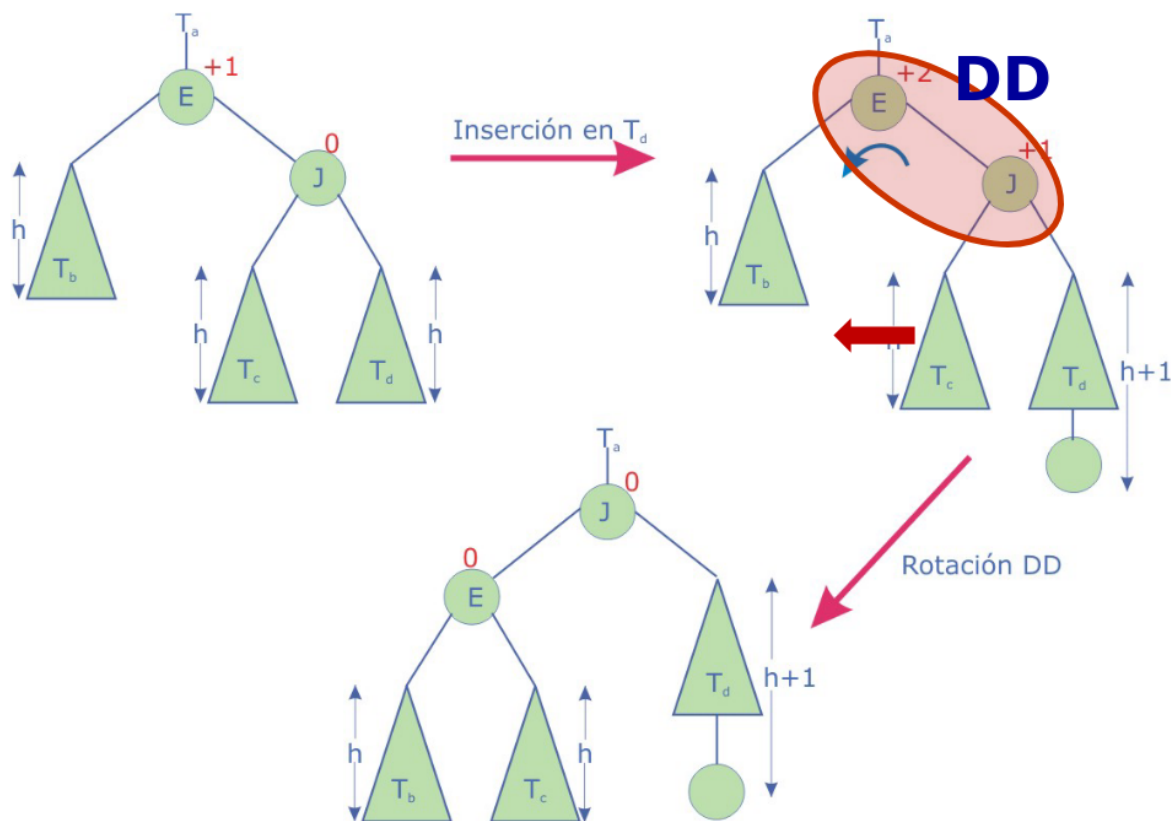


Figure 3: archivos/imagenes/Pasted image 20241003202659.png

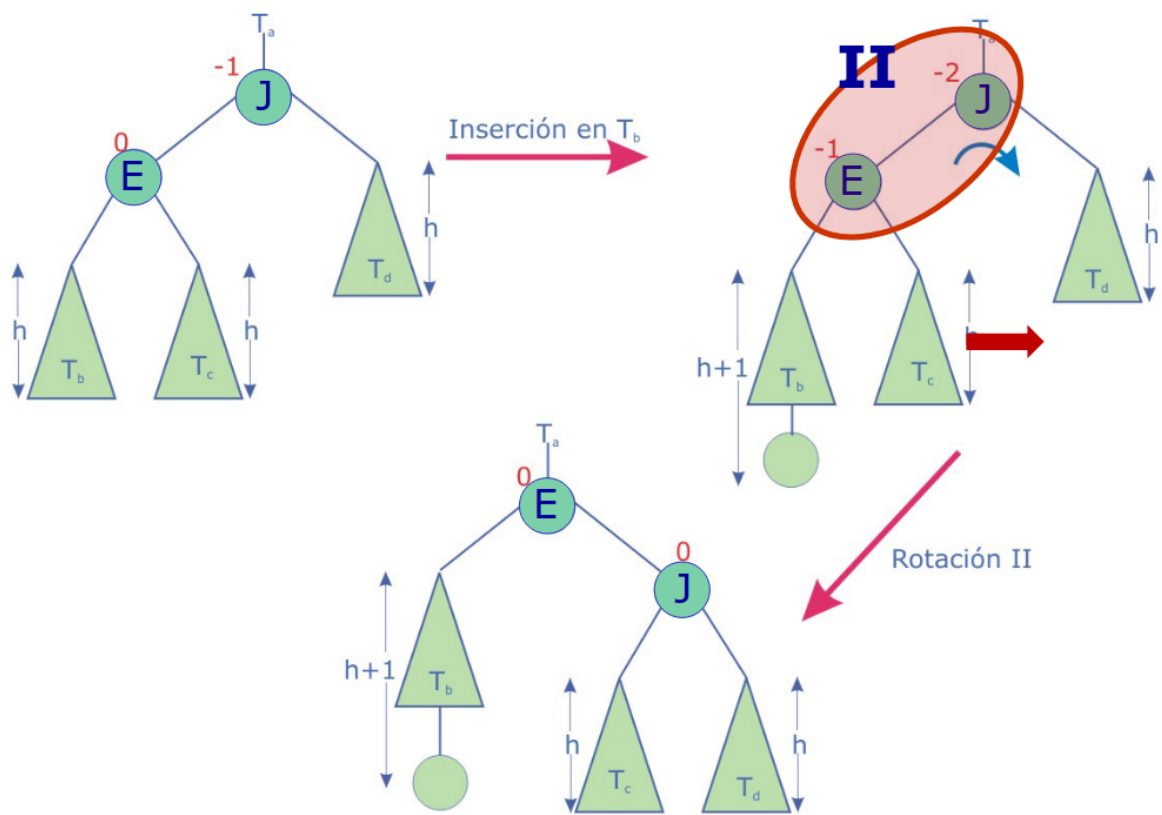


Figure 4: archivos/imagenes/Pasted image 20241003202732.png

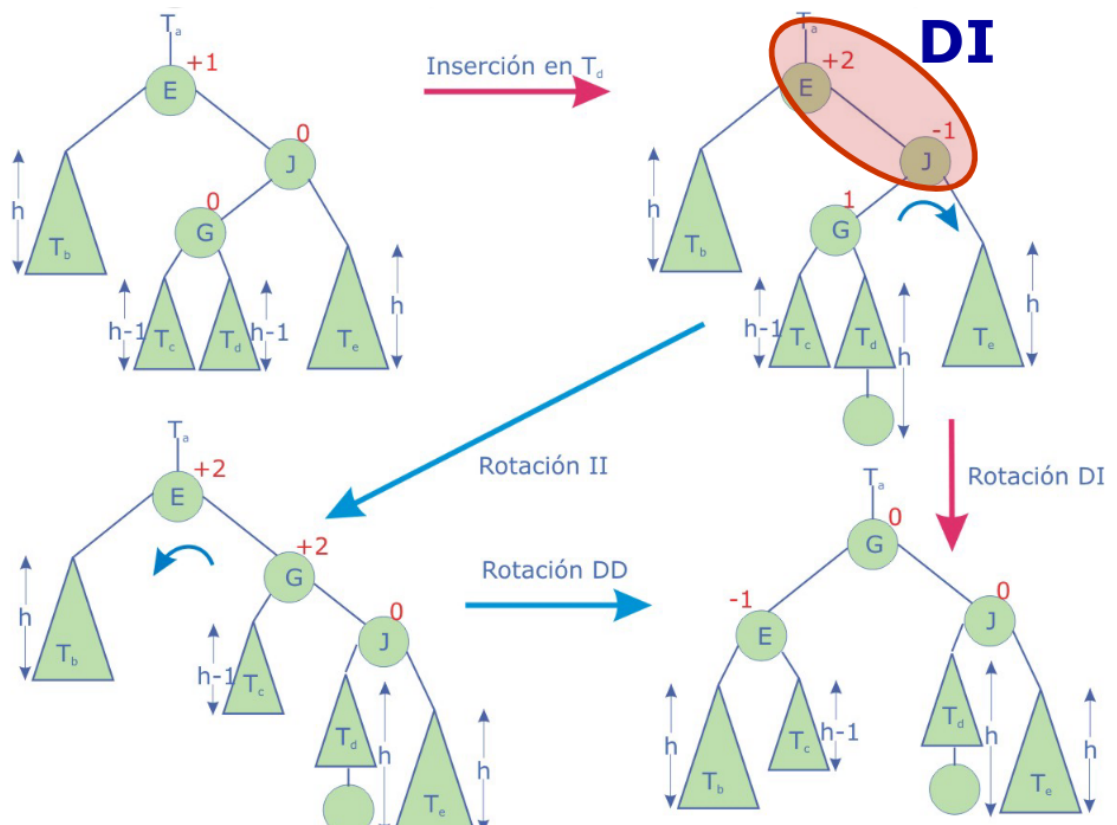
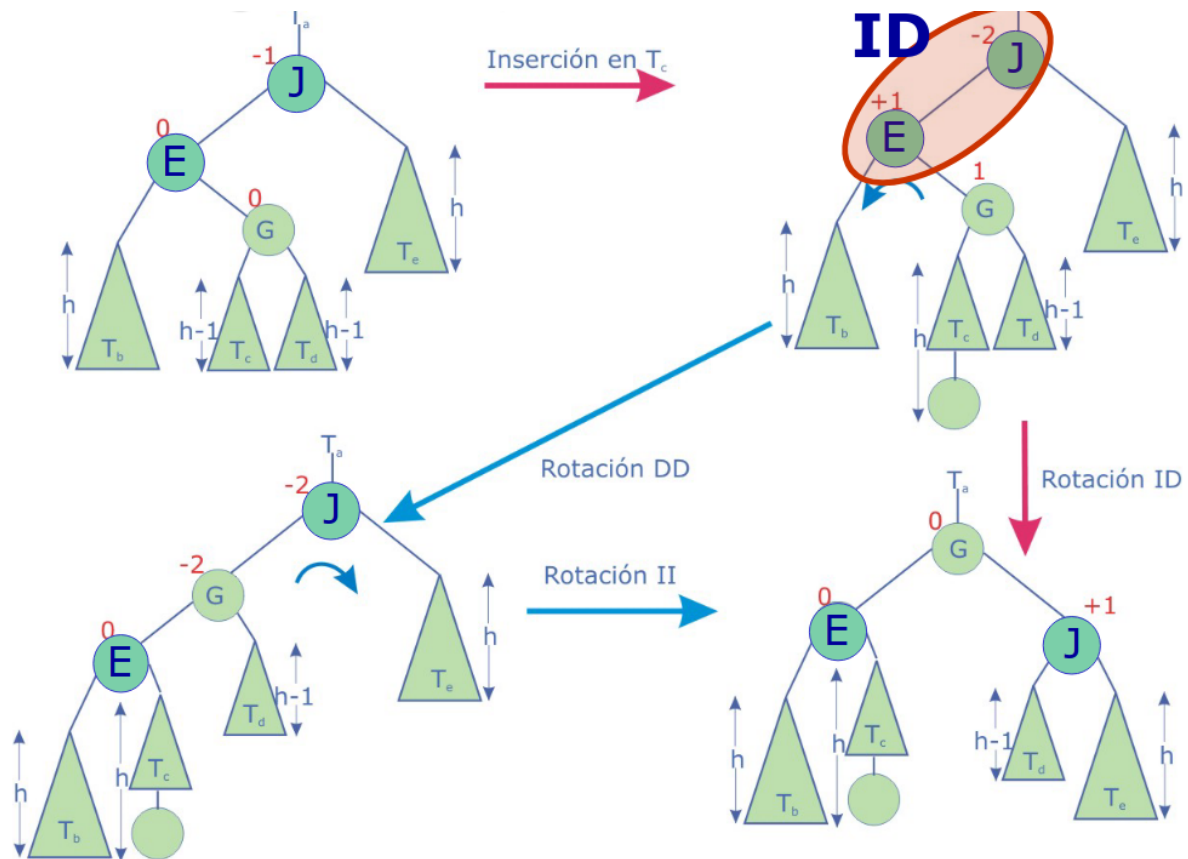
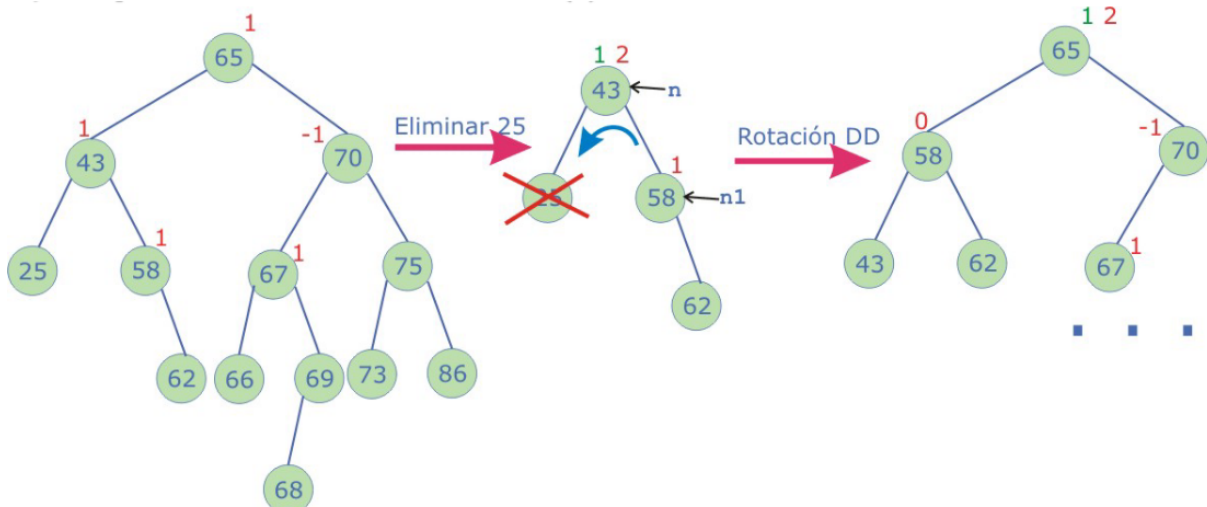


Figure 5: archivos/imagenes/Pasted image 20241003202813.png

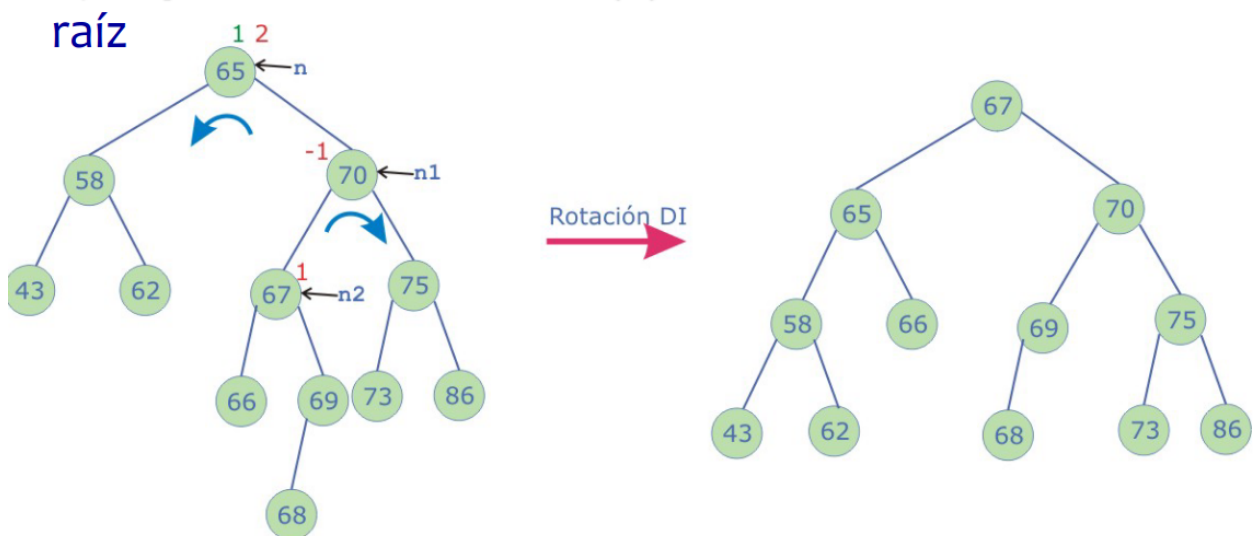
Rotaciones Complicadas



Eliminación Se sigue el **mismo algoritmo que en ABB**, pero **incluyendo las reestructuraciones necesarias**: - Si el nodo es **hoja**, se suprime - Si solo tiene **un descendiente**, se sustituye por su descendiente y se elimina. - Si tiene **2 subárboles**, se busca el nodo más a la derecha del **subárbol derecho**, o el nodo más a la izquierda del subárbol derecho, se sustituye, se elimina y se comprueba si hace falta reestructurar.



- 25 es una hoja, simplemente se elimina
- Vuelvo por el camino hasta la raíz viendo si hay desequilibrios:
 - $fe(43)=+2 \Rightarrow$ Miro subárbol derecho
 - $fe(HD)=fe(58) \geq 0 \Rightarrow$ Rotación DD
- **SIGO HASTA LA RAÍZ, PARA COMPROBAR SI HAY MÁS DESEQUILIBRIOS**



- **SIGO HASTA LA RAÍZ, PARA COMPROBAR SI HAY MÁS DESEQUILIBRIOS**
- Vuelvo por el camino hasta la raíz viendo si hay desequilibrios:
 - $fe(65)=+2 \Rightarrow$ Miro subárbol derecho
 - $fe(HD)=fe(70) < 0 \Rightarrow$ Rotación DI