

# Árboles balanceados (AVL)

Estructura de Datos

# Árbol Balanceado (AVL)

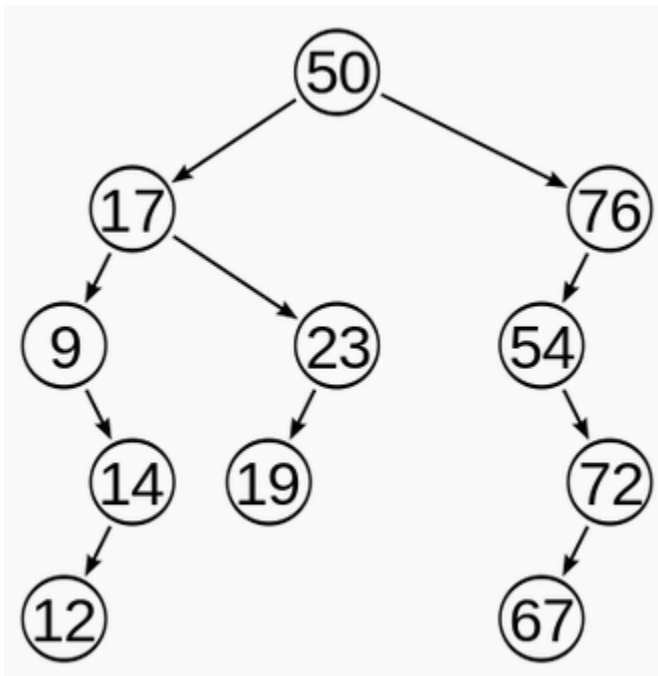
- ▶ Un árbol AVL es un tipo especial de árbol binario ideado por los matemáticos rusos Adelson-Velskii y Landis.
- ▶ Fue el primer árbol de búsqueda binario auto-balanceable que se ideó.
- ▶ Los árboles AVL están siempre equilibrados de tal modo que para todos los nodos, la altura de la rama izquierda no difiere en más de una unidad de la altura de la rama derecha o viceversa.
- ▶ Gracias a esta forma de equilibrio (o balanceo), la complejidad de una búsqueda en uno de estos árboles se mantiene siempre en orden de complejidad  $O(\log n)$ .

# Árbol Balanceado (AVL)

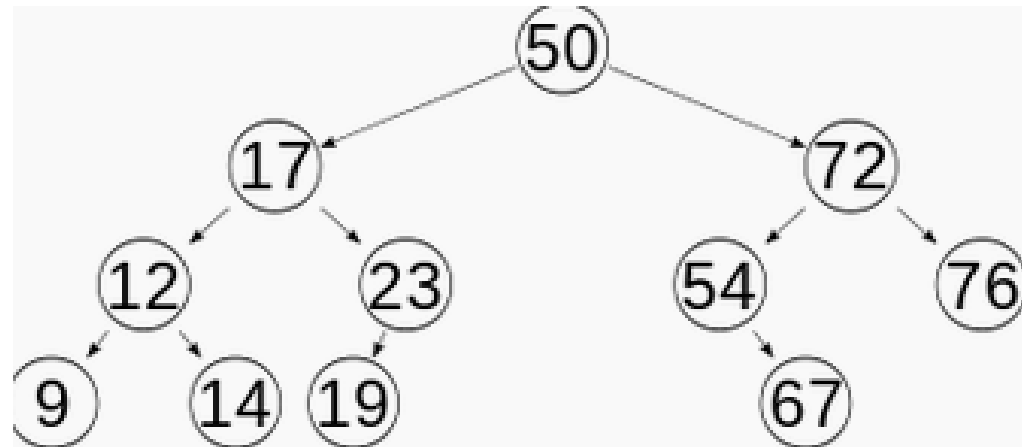
- ▶ El factor de equilibrio puede ser almacenado directamente en cada nodo o ser computado a partir de las alturas de los subárboles.
- ▶ Para conseguir esta propiedad de equilibrio, la inserción y el borrado de los nodos se ha de realizar de una forma especial.
- ▶ Si al realizar una operación de inserción o borrado se rompe la condición de equilibrio, hay que realizar una serie de rotaciones de los nodos.

# Ejemplo de árbol Balanceado (AVL)

Árbol no equilibrado (no es AVL)



Árbol equilibrado (sí es AVL)



# Definición formal

## ► Definición de la altura de un árbol

Sea  $T$  un árbol binario de búsqueda y sean  $T_i$  y  $T_d$  sus subárboles, su altura  $H(T)$ , es:

- 0 si el árbol  $T$  contiene solo la raíz
- $1 + \max(H(T_i), H(T_d))$  si contiene más nodos

## ► Definición de árbol AVL

- Un árbol vacío es un árbol AVL
- Si  $T$  es un árbol no vacío y  $T_i$  y  $T_d$  sus subárboles, entonces  $T$  es AVL si y solo si:
  - $T_i$  es AVL
  - $T_d$  es AVL
  - $|H(T_i) - H(T_d)| \leq 1$

# Factor de equilibrio

- ▶ Cada nodo, además de la información que se pretende almacenar, debe tener los dos punteros a los árboles derecho e izquierdo, igual que los árboles binarios de búsqueda (ABB), y además el dato que controla el factor de equilibrio.
- ▶ El factor de equilibrio es la diferencia entre las alturas del árbol derecho y el izquierdo:
  - ▶  $FE = \text{altura subárbol derecho} - \text{altura subárbol izquierdo}$
  - ▶ Por definición, para un árbol AVL, este valor debe ser -1, 0 ó 1.
- ▶ Si el factor de equilibrio de un nodo es:
  - ▶ 0 -> el nodo está equilibrado y sus subárboles tienen exactamente la misma altura.
  - ▶ 1 -> el nodo está equilibrado y su subárbol derecho es un nivel más alto.
  - ▶ -1 -> el nodo está equilibrado y su subárbol izquierdo es un nivel más alto.
- ▶ Si el factor de equilibrio  $|FE| \geq 2$  es necesario reequilibrar.

# Operaciones

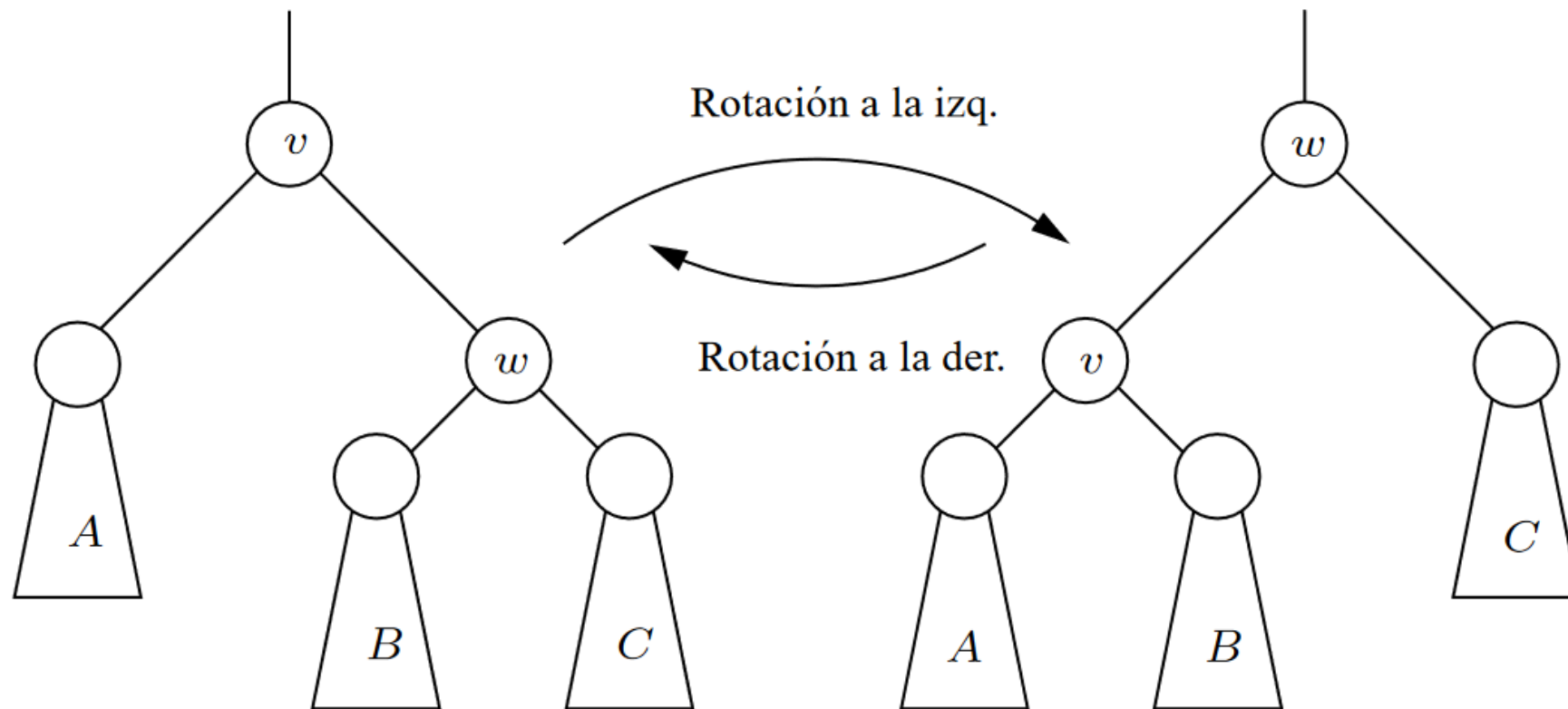
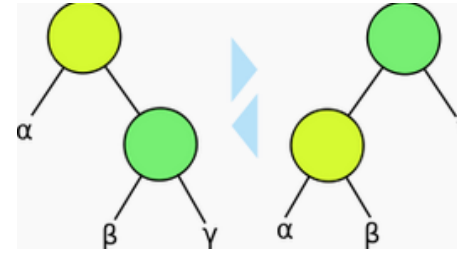
- ▶ Las operaciones básicas de un árbol AVL implican generalmente el realizar los mismos algoritmos que serían realizados en un árbol binario de búsqueda desequilibrado, pero precedido o seguido por una o más de las llamadas "rotaciones AVL".

## Rotaciones

- ▶ El reequilibrado se produce de abajo hacia arriba sobre los nodos en los que se produce el desequilibrio.
- ▶ Pueden darse dos casos:
  - ▶ Rotación simple o rotación doble; a su vez ambos casos pueden ser hacia la derecha o hacia la izquierda.

# Operaciones

## Rotaciones





# Operaciones

## Rotaciones

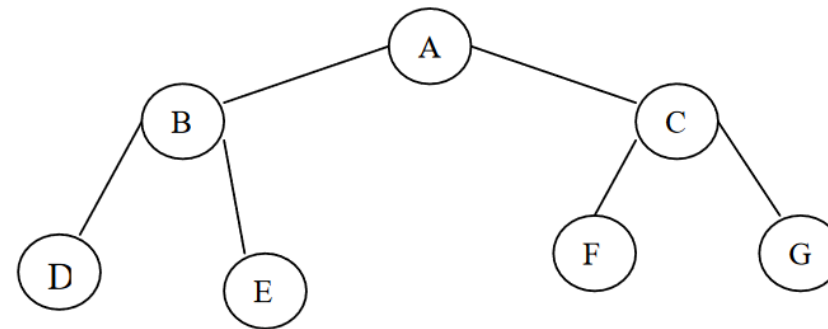
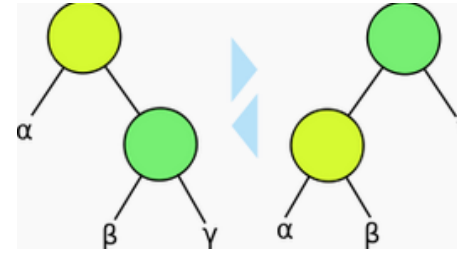
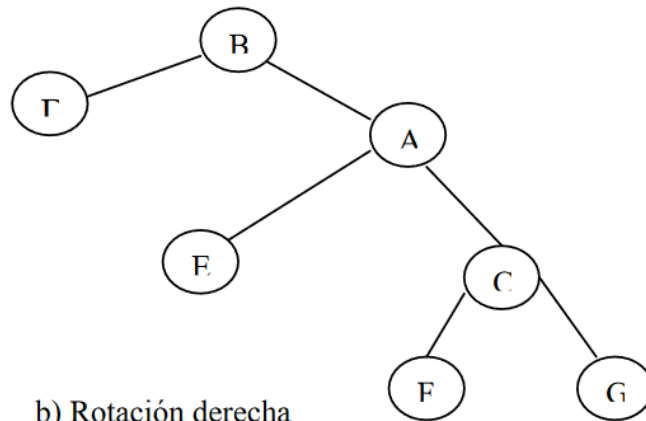
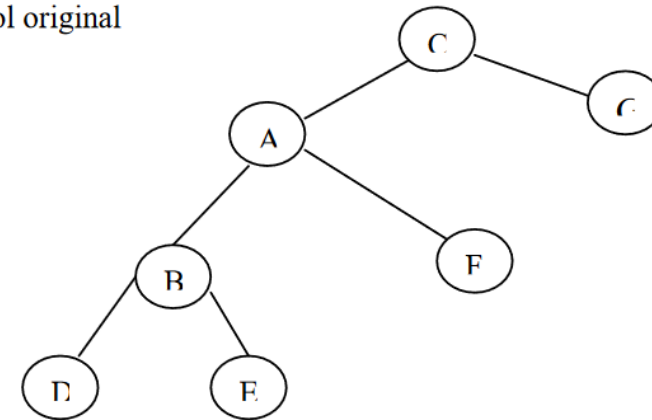


Figura 1

a) Árbol original



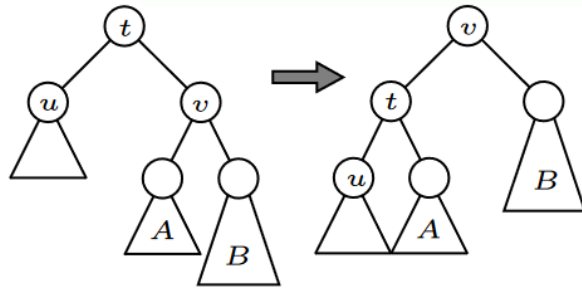
b) Rotación derecha



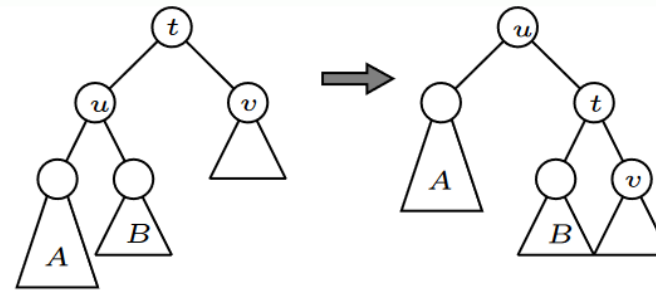
c) Rotación izquierda

# Operaciones

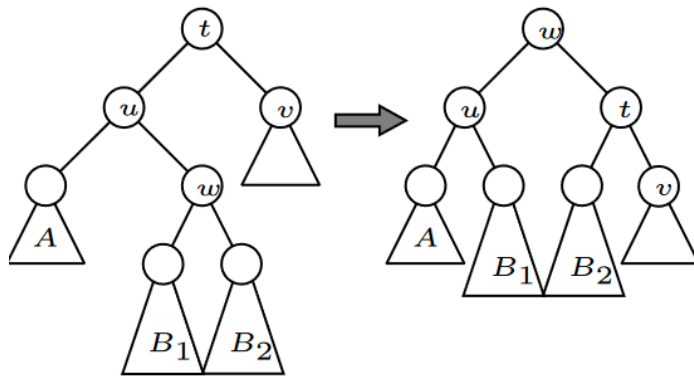
## Rotaciones



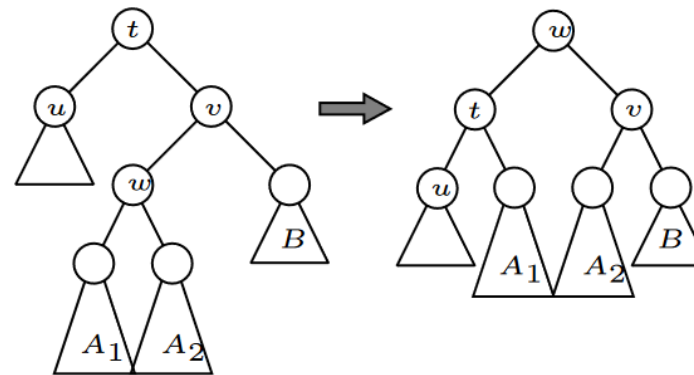
Rotación simple izquierda



Rotación simple derecha



Rotación doble izquierda-derecha

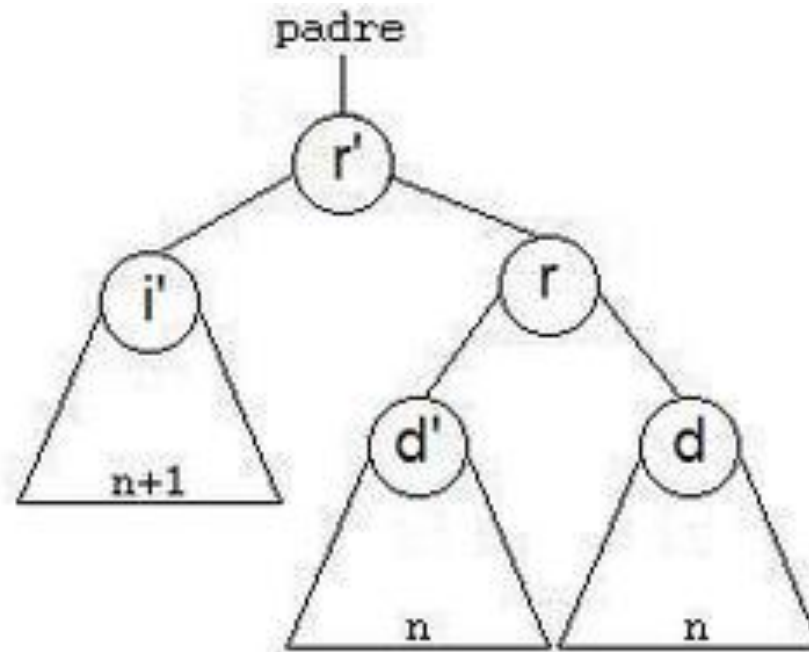
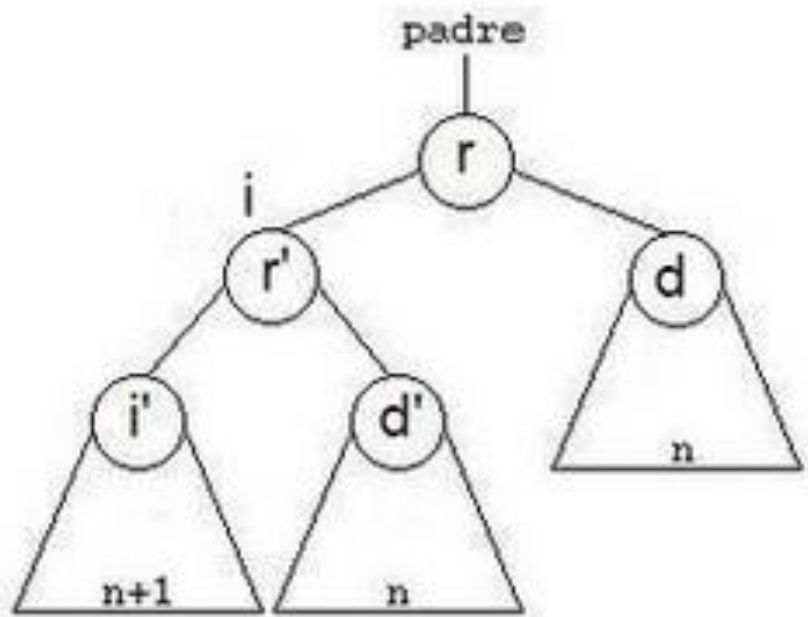


Rotación doble derecha-izquierda

# Rotación simple a la derecha

- ▶ De un árbol de raíz (r) y de hijos izquierdo (i) y derecho (d)
- ▶ Formar un nuevo árbol cuya raíz sea la raíz del hijo izquierdo
- ▶ Como hijo izquierdo se coloca el hijo izquierdo de i (i')
- ▶ Como hijo derecho se construye un nuevo árbol que tendrá como raíz, la raíz del árbol (r)
- ▶ El hijo derecho de i (d') será el hijo izquierdo
- ▶ El hijo derecho será el hijo derecho del árbol (d).

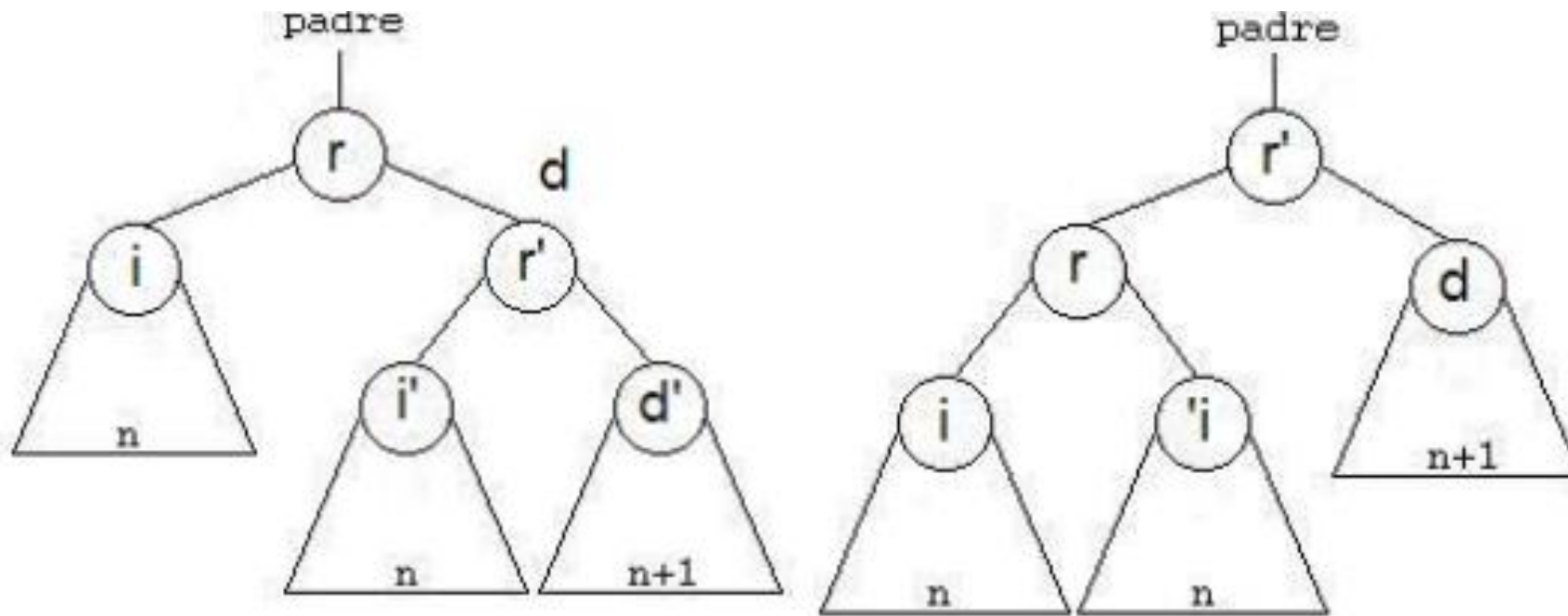
# Rotación simple a la derecha



# Rotación simple a la izquierda

- ▶ De un árbol de raíz (r) y de hijos izquierdo (i) y derecho (d)
- ▶ Consiste en formar un nuevo árbol cuya raíz sea la raíz del hijo derecho
- ▶ Como hijo derecho se coloca el hijo derecho de d (d')
- ▶ Como hijo izquierdo se construye un nuevo árbol que tendrá como raíz la raíz del árbol (r)
- ▶ El hijo izquierdo de d será el hijo derecho (i')
- ▶ El hijo izquierdo será el hijo izquierdo del árbol (i).
- ▶ Precondición : Tiene que tener hijo derecho no vacío.

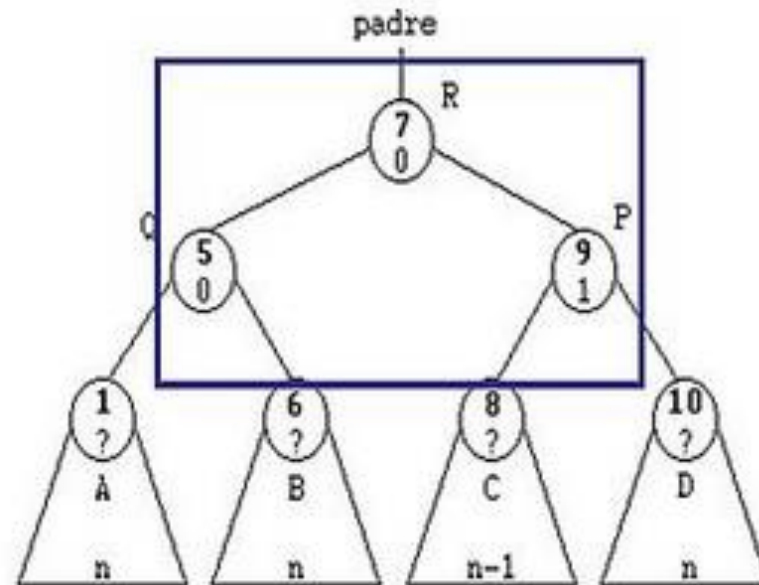
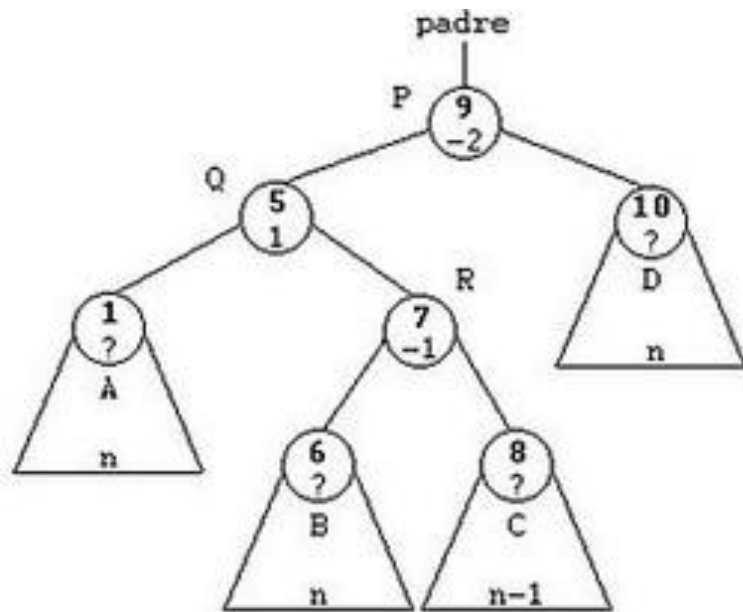
# Rotación simple a la izquierda



- Si la inserción se produce en el hijo derecho del hijo izquierdo del nodo desequilibrado (o viceversa) hay que realizar una doble rotación.

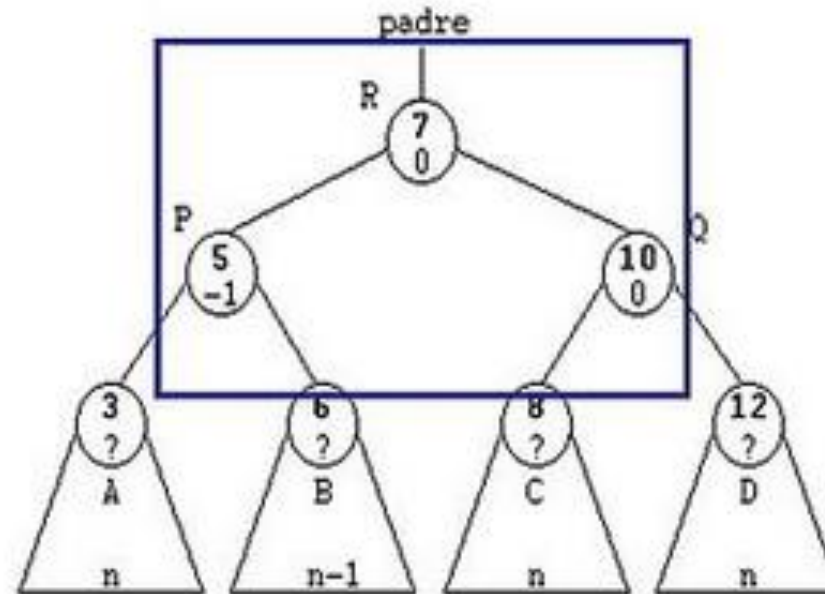
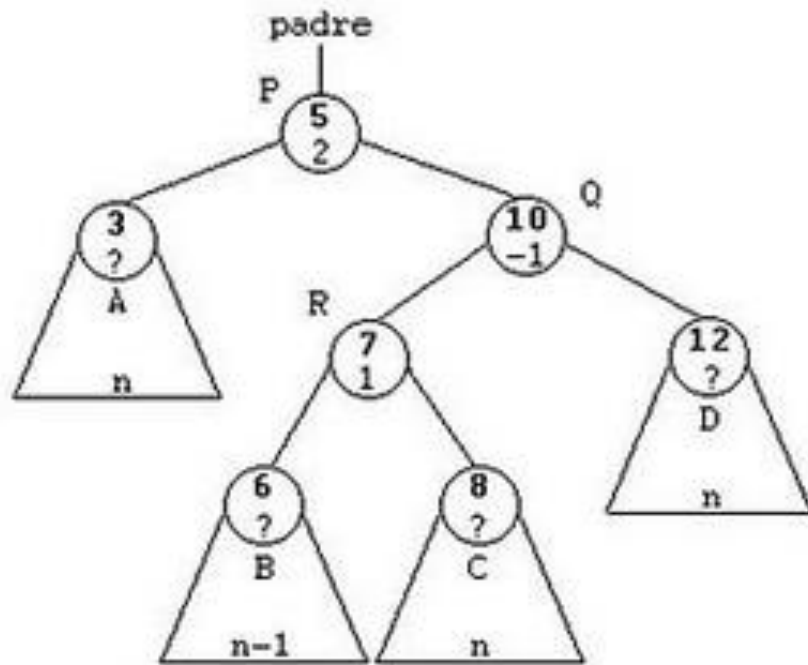
# Rotación doble a la derecha

- La rotación doble a la derecha son dos rotaciones simples, primero rotación simple izquierda y luego rotación simple derecha.



# Rotación doble a la izquierda

- La Rotación doble a la Izquierda son dos rotaciones simples, primero rotación simple derecha y luego rotación simple izquierda.

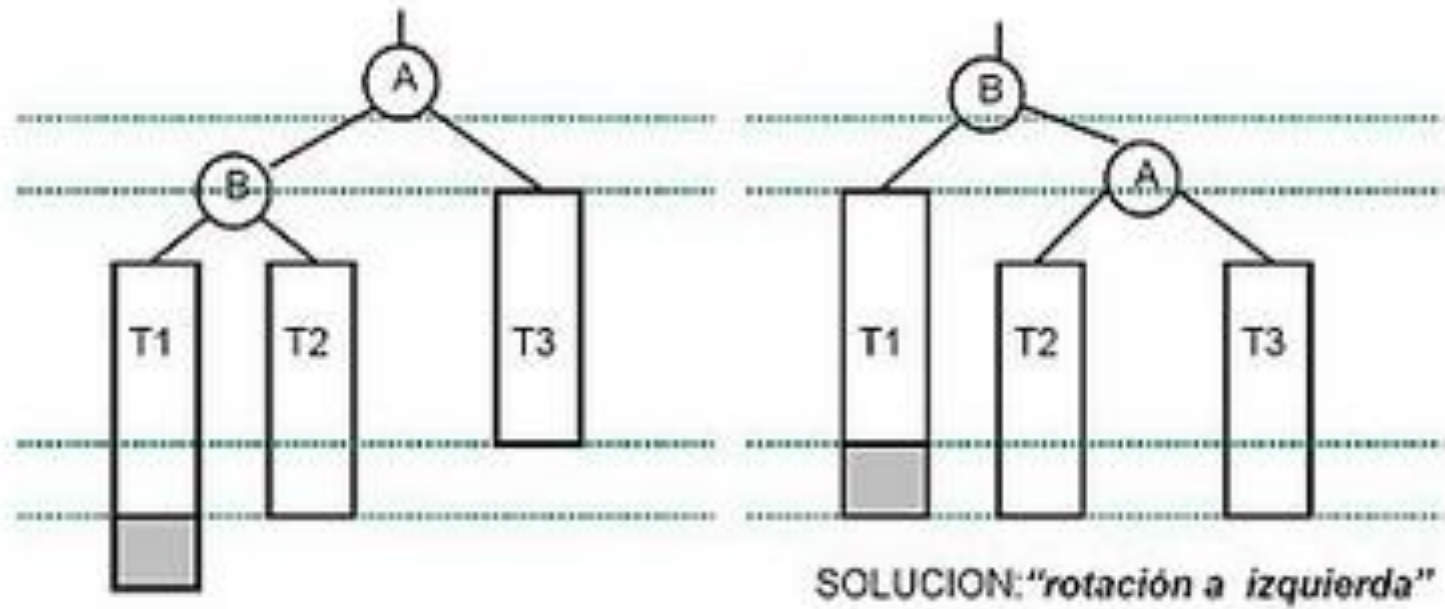




# Inserción

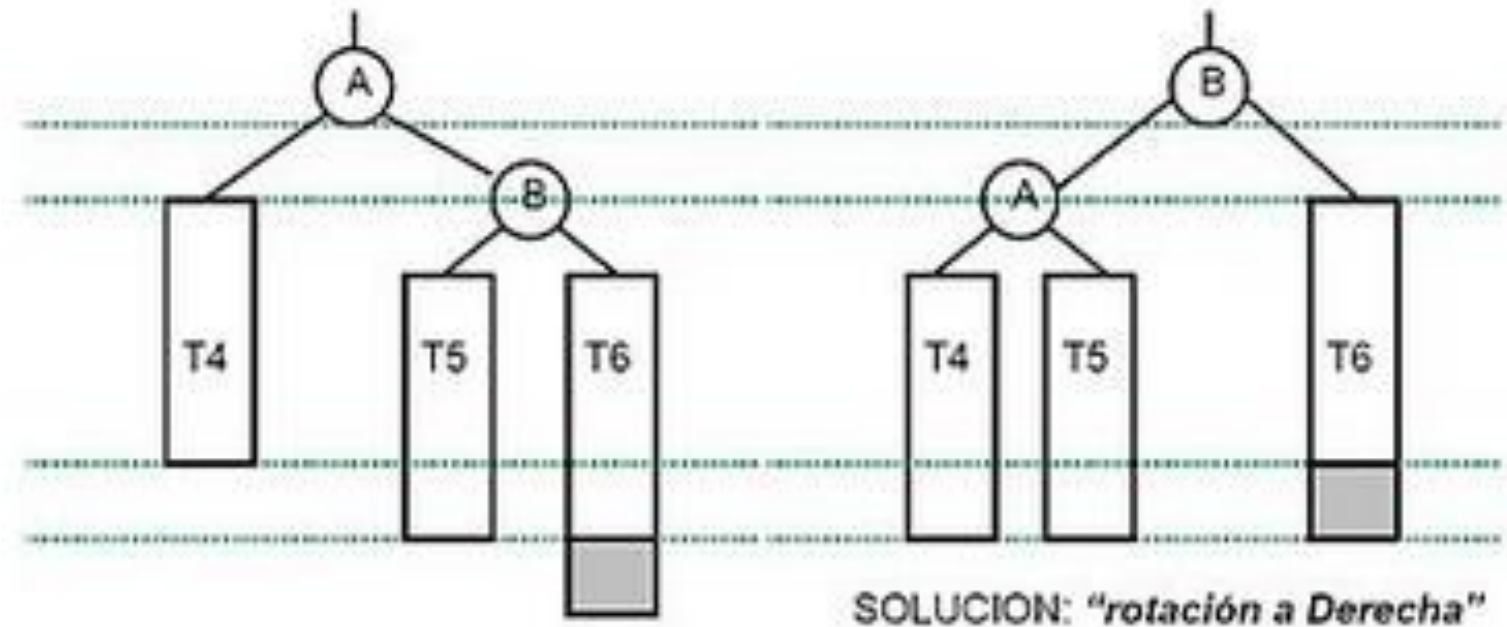
- ▶ La inserción en un árbol de AVL puede ser realizada insertando el valor dado en el árbol como si fuera un árbol de búsqueda binario desequilibrado y después retrocediendo hacia la raíz, rotando sobre cualquier nodo que pueda haberse desequilibrado durante la inserción.
- ▶ Proceso de inserción:
  - ▶ Buscar hasta encontrar la posición de inserción o modificación (proceso idéntico a inserción en árbol binario de búsqueda)
  - ▶ Insertar el nuevo nodo con factor de equilibrio “equilibrado”
  - ▶ Desandar el camino de búsqueda, verificando el equilibrio de los nodos, y re-equilibrando si es necesario

# Inserción



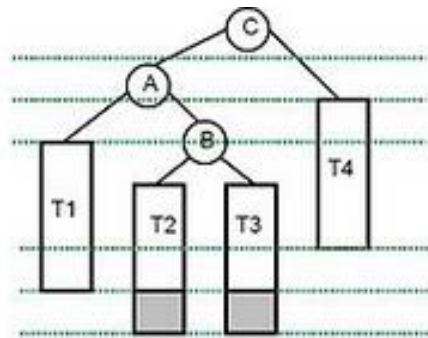
**Caso 1: Izquierda-Izquierda**

# Inserción

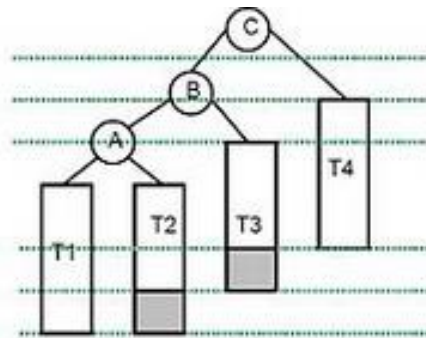


**Caso 2: Derecha- Derecha**

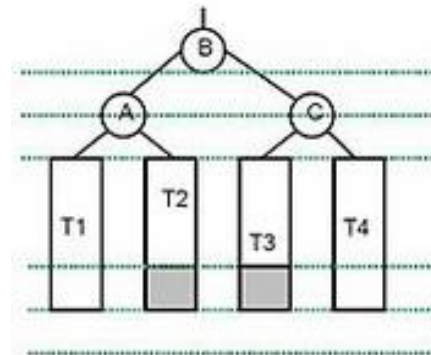
# Inserción



Caso 3: Izq-Dech

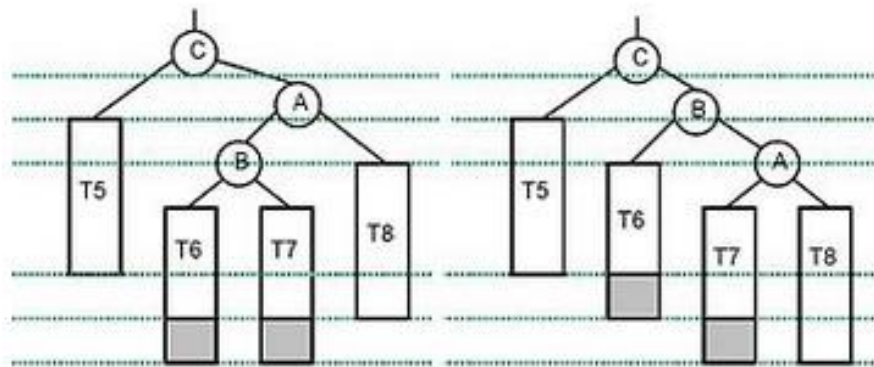


SOLUCION: "Rotación Doble"  
PASO 1: "rotación a derecha"



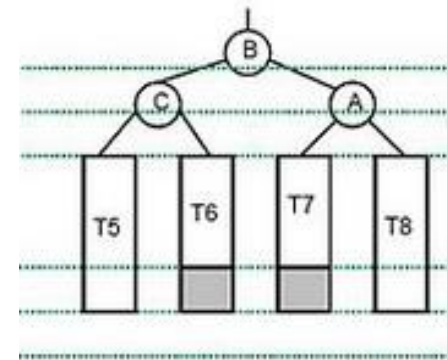
SOLUCION: "Rotación Doble"  
PASO 2: "rotación a izquierda"

# Inserción



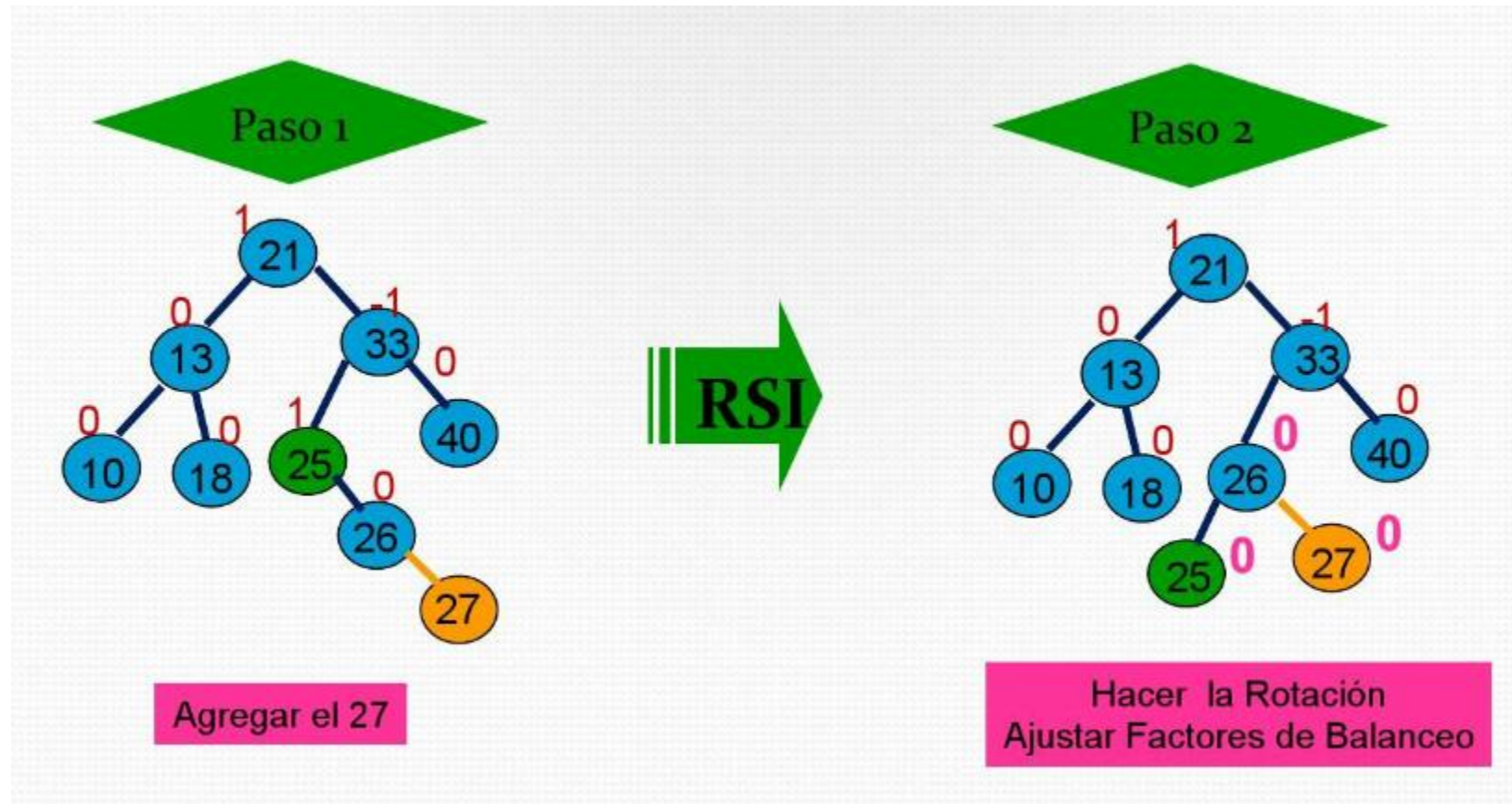
Caso 4: Dech-Izq

SOLUCION: "Rotación Doble"  
PASO 1: "rotación a izquierda"



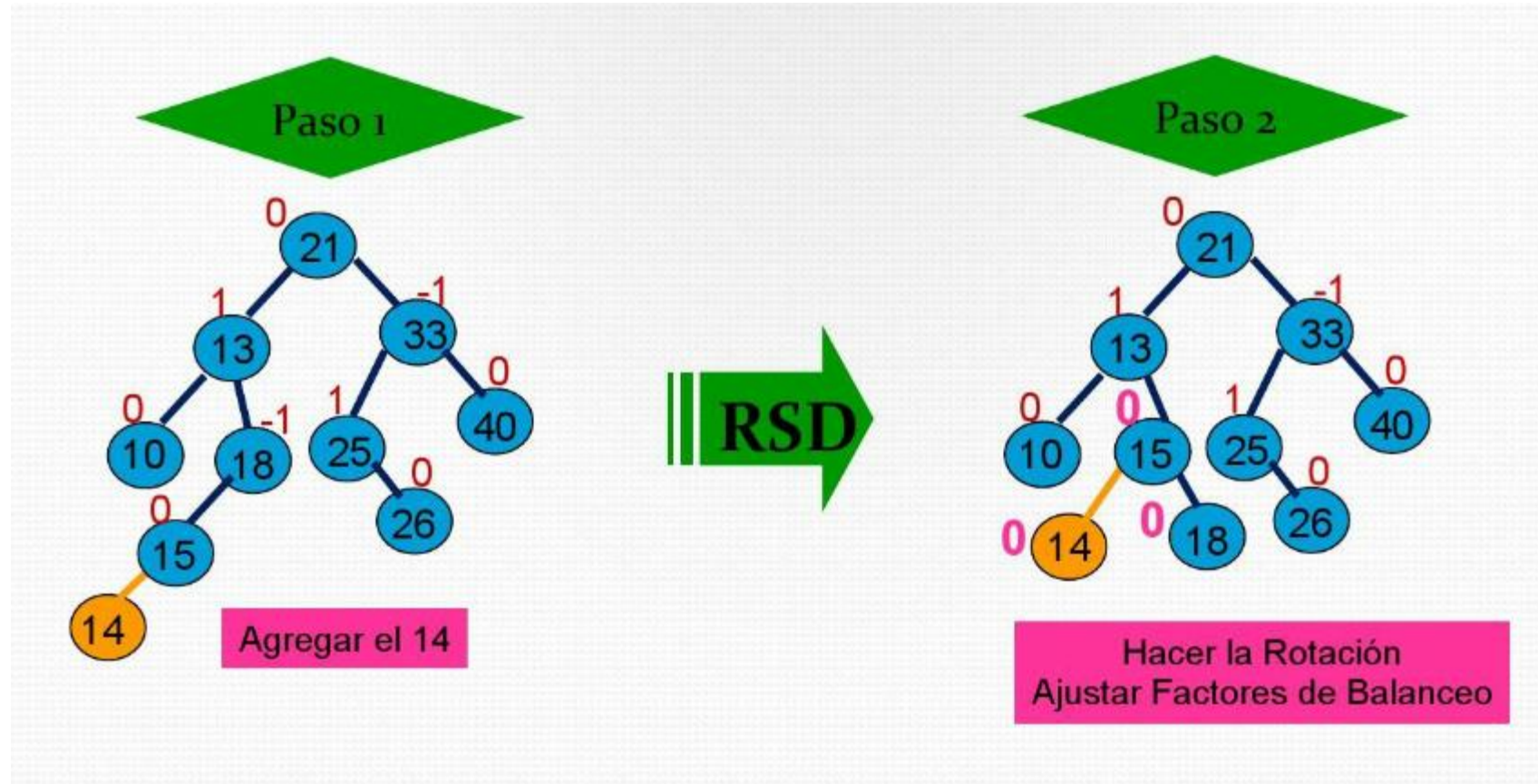
SOLUCION: "Rotación Doble"  
PASO 2: "rotación a derecha"

# Inserción





# Inserción

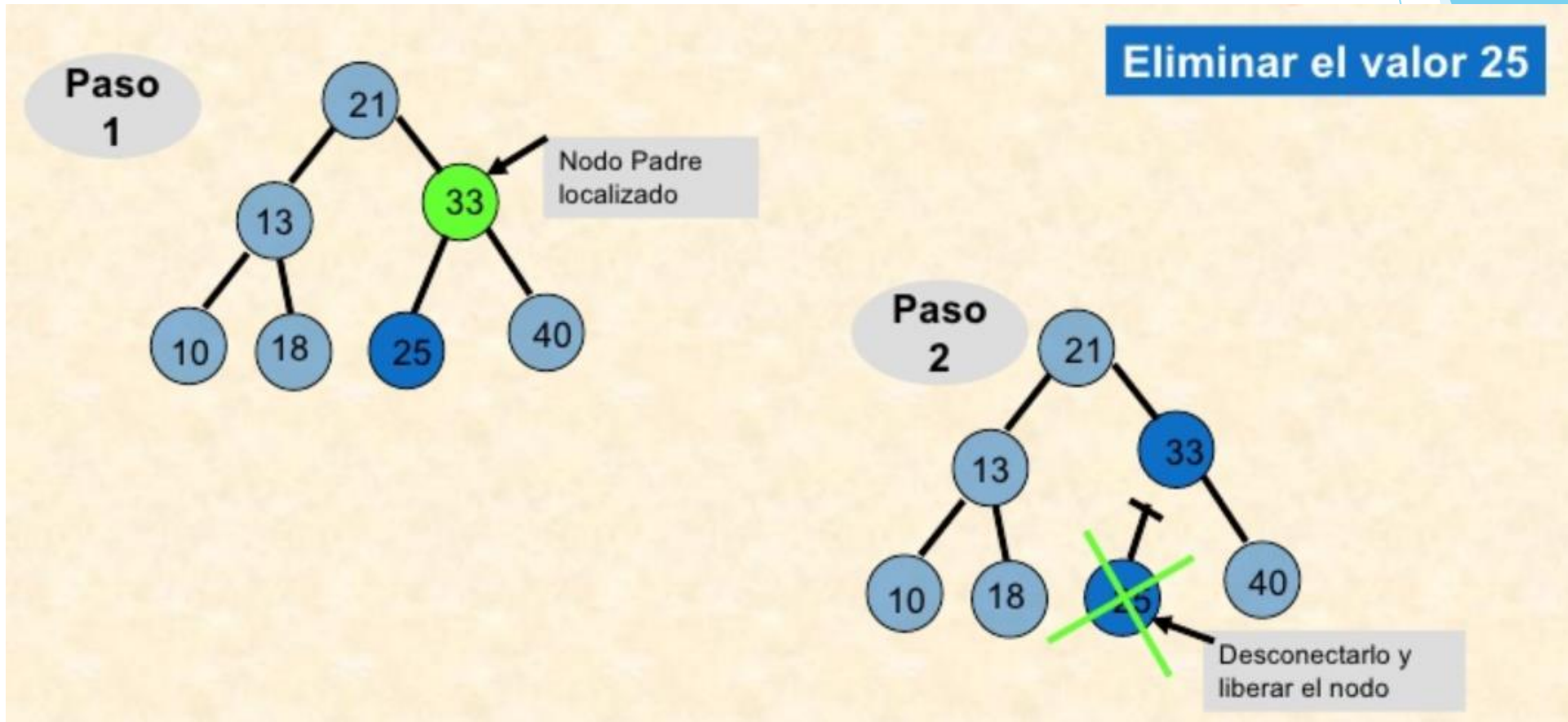


# Extracción

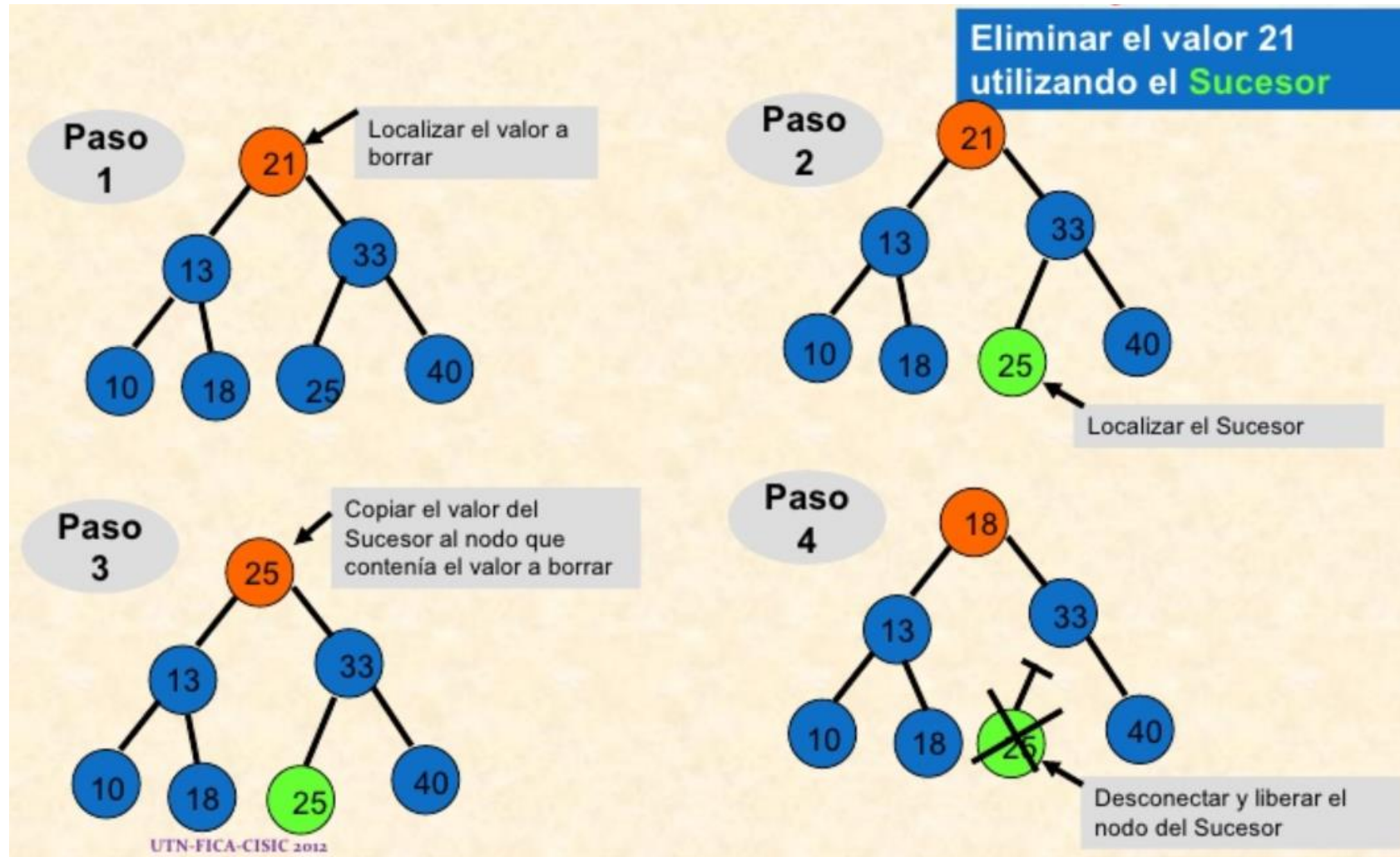
- ▶ El procedimiento de borrado es el mismo que en el caso de árbol binario de búsqueda.
- ▶ La diferencia se encuentra en el proceso de reequilibrado posterior.
- ▶ El problema de la extracción puede resolverse en  $O(\log(n))$  pasos.
- ▶ Una extracción trae consigo una disminución de la altura de la rama donde se extrajo y tendrá como efecto un cambio en el factor de equilibrio del nodo padre de la rama en cuestión, pudiendo necesitarse una rotación.
- ▶ Esta disminución de la altura y la corrección de los factores de equilibrio con sus posibles rotaciones asociadas pueden propagarse hasta la raíz.



# Extracción

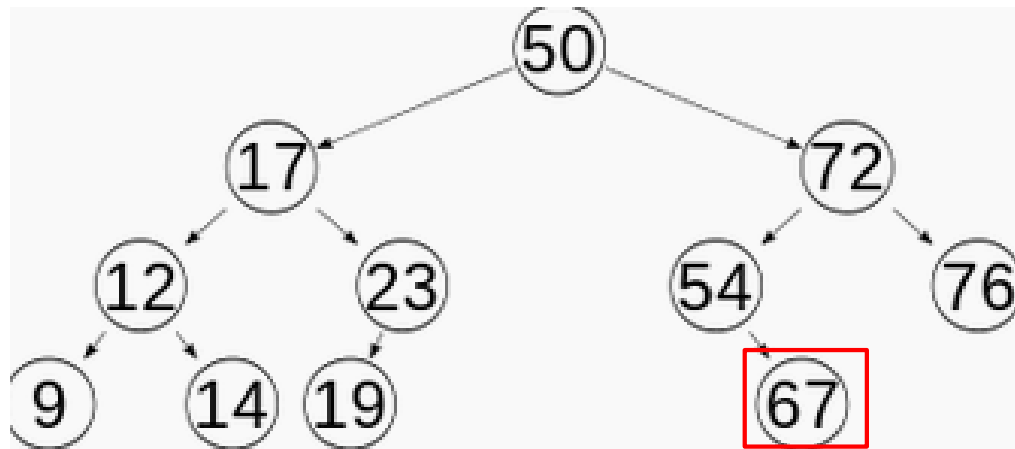


# Extracción



# Búsqueda

- ▶ El procedimiento de búsqueda es el mismo que en el caso de árbol binario de búsqueda.
- ▶ Ejemplo:
  - ▶ Buscar el nodo 67



**GRACIAS**