

Arboles Binarios Balanceados

Estructuras de Datos

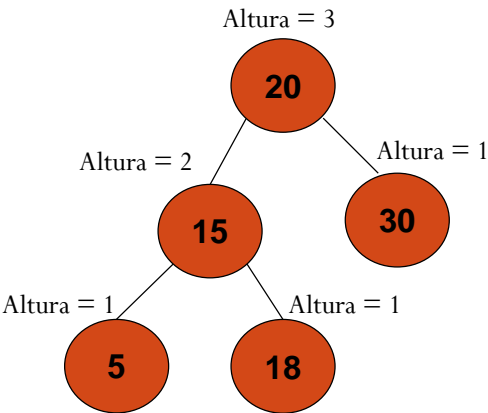
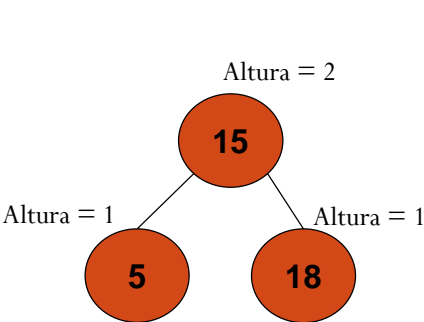
1

- Árboles balanceados
 - Por altura (AVL)
 - Por peso (perfectamente balanceados)
 - Rojinegros

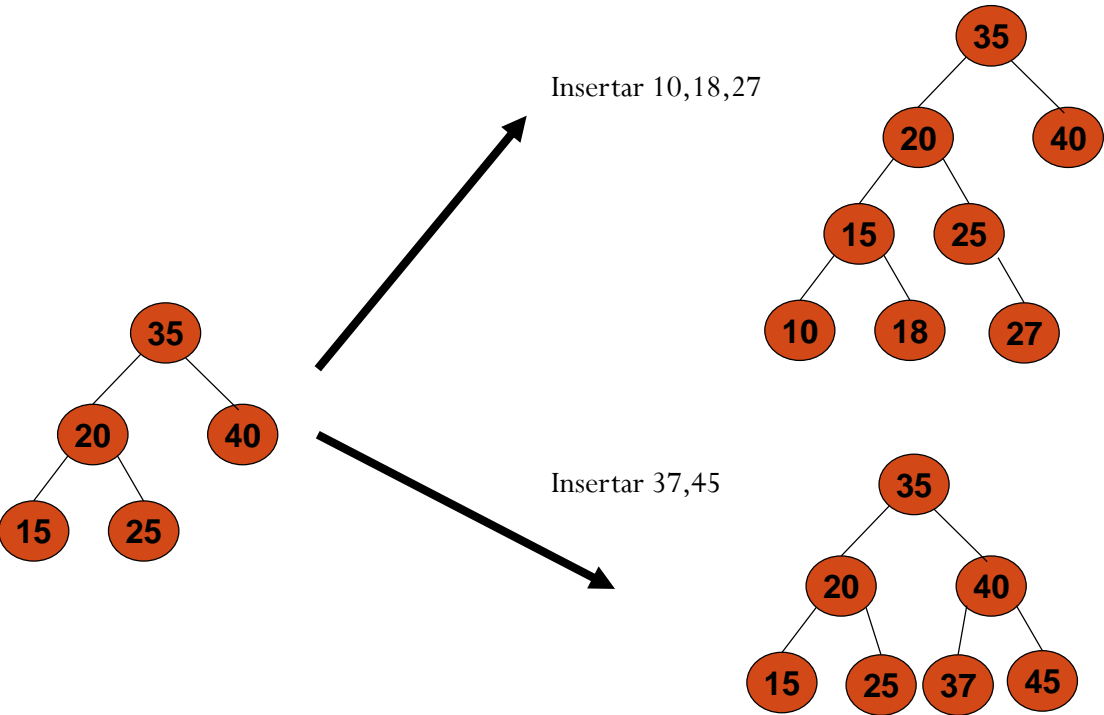
Árboles Balanceados

- Árboles balanceados son aquellos ABB que cumplen con una condición de equilibrio (que sus subárboles izquierdo y derecho tengan la misma profundidad)
- Los árboles balanceados optimizan la búsqueda de elementos.
- Árboles AVL (1962- Adelson, Velskii y Landis) la altura de los subárboles asociados a cada elemento no pueden diferir en más de 1 y los dos subárboles son también AVL.

Ejemplos de árboles AVL



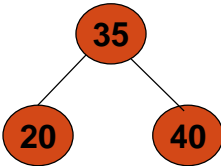
Inserción en árboles balanceados



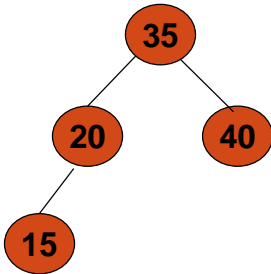
Inserción en árboles balanceados

Caso 1.

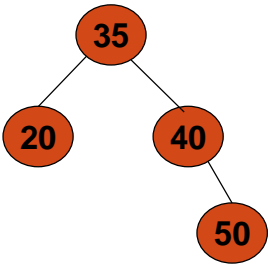
Las ramas izquierda y derecha del árbol tienen la misma altura.
($H_{RI} = H_{RD}$)



Caso 1.1 Inserta elemento en rama izquierda (RI)



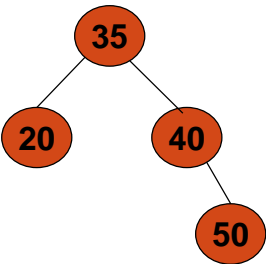
Caso 1.2 Inserta elemento en rama derecha (RD)



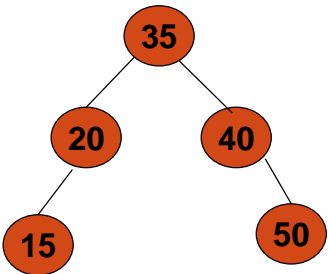
Inserción en árboles balanceados

Caso 2.
Las ramas izquierda y derecha del árbol tienen altura diferente.
($H_{RI} \neq H_{RD}$)

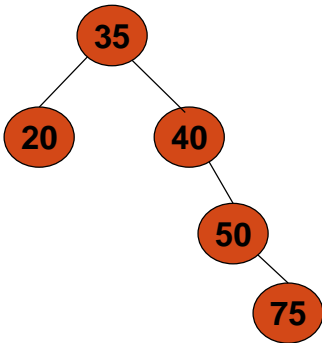
Caso 2.1 suponer $H_{RI} < H_{RD}$



Caso 2.1.1 Inserta en RI



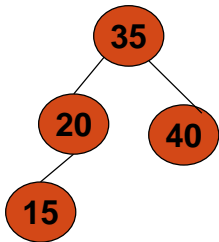
Caso 2.1.2 Inserta en RD



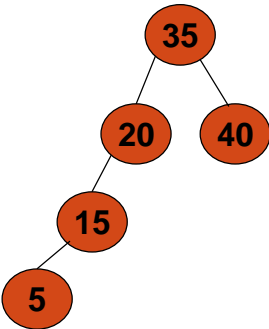
Inserción en árboles balanceados

Caso 2. (continuación)
Las ramas izquierda y derecha del árbol tienen altura diferente.
($H_{RI} \neq H_{RD}$)

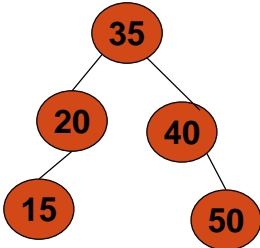
Caso 2.2 suponer $H_{RI} > H_{RD}$



Caso 2.2.1 Inserta en RI

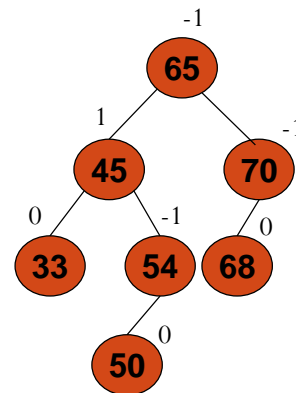


Caso 2.1.2 Inserta en RD



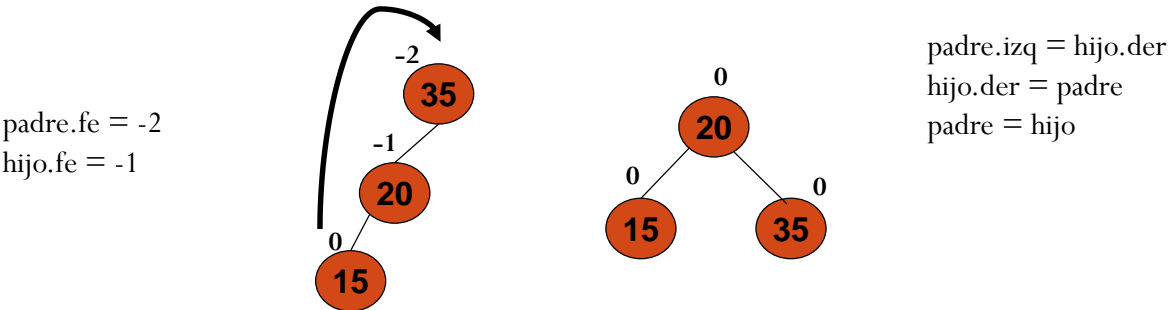
Reestructuración

- **Factor de equilibrio:** Es la diferencia entre la altura de la rama derecha y la altura de la rama izquierda. $FE = H_{RD} - H_{RI}$
- Cada nodo tiene asociado un factor de equilibrio que se calcula para todos los elementos de la rama donde se realizó la inserción. Este se utiliza para saber si el árbol esta balanceado o debe reestructurarse.

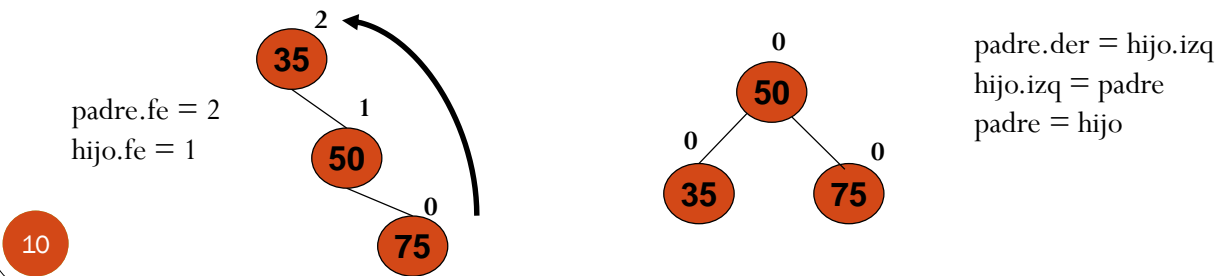


Rotaciones

Caso 1 Rotación simple por la rama izquierda

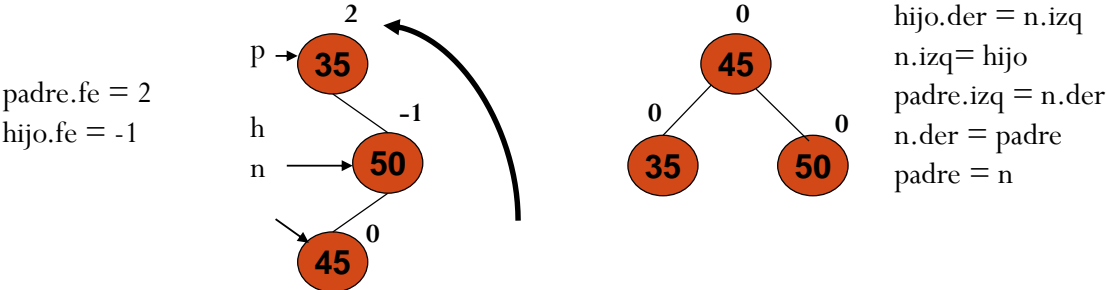


Caso 2 Rotación simple por la rama derecha

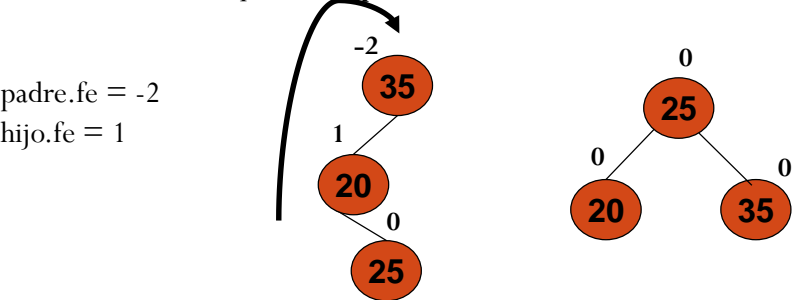


Rotaciones

Caso 3 Rotación compuesta Derecha Izquierda



Caso 4 Rotación compuesta Izquierda Derecha



Recordemos los Arboles Binarios

- ¿Cuáles fueron sus ventajas?
- Podemos realizar las siguientes operaciones rápidamente:
 - Inserción
 - Eliminación
 - Búsqueda
- Porque para n nodos:
 - La altura de un árbol binario es $\log(n)+1$
 - Cada operación requiere iterativamente o recursivamente el chequeo de los nodos hijos

Arboles binarios no balanceados

- Formemos dos arboles de búsqueda binaria
- Uno insertando la siguiente secuencia:
 - 10 20 30 40 50 60 70 80 90 100
- Otro insertando la siguiente secuencia:
 - 100 90 80 70 60 50 40 30 20 10
- ¿Qué sucede? ¿Aun podemos realizar la inserción, eliminación y búsqueda de forma rápida?
- ¿Cual es el problema fundamental?

El concepto de rotacion (ignore el sombreado)

- La rotacion involucra tres nodos:
 - Un padre y sus dos hijos
- Para una rotacion derecha, el hijo izquierdo del nodo pasa a ser el padre:



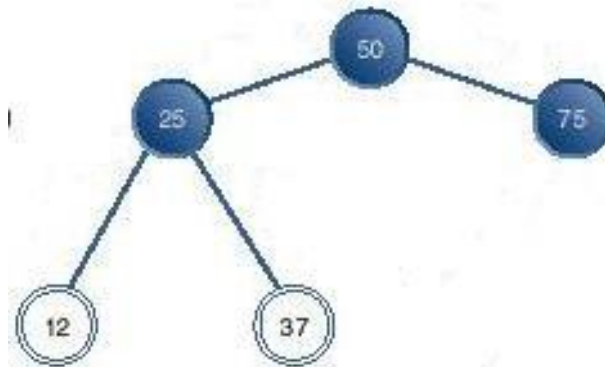
- La rotacion izquierda es simplemente lo opuesto.
Necesitaremos de estas operaciones.

Las rotaciones ayudan a balancear el árbol

- Por ejemplo, si el lado izquierdo de un árbol es "pesado", haciendo una rotación hacia la derecha ayudaría a equilibrar las cosas
- Las rotaciones, en general:
 - Levante algunos nodos y baje otros para ayudar a equilibrar el árbol
 - Asegúrese de no violar ninguna de las características de un árbol de búsqueda binaria
 - Por lo tanto todos los nodos a la izquierda todavía deben tener valores más pequeños
 - Todos los nodos de la derecha aún deben tener valores mayores

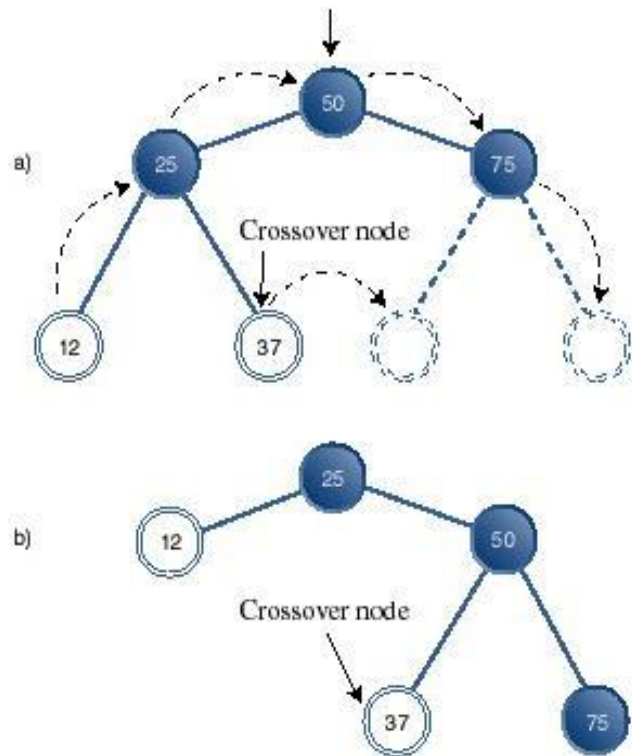
La rotaciones involucran muchos nodos

- Una rotación de tres nodos fue fácil.
- Veamos una más complicada
- Digamos que queremos rotar el siguiente árbol, con 50 como nuestro "nodo superior". ¿Qué nodo se convierte en la raíz?



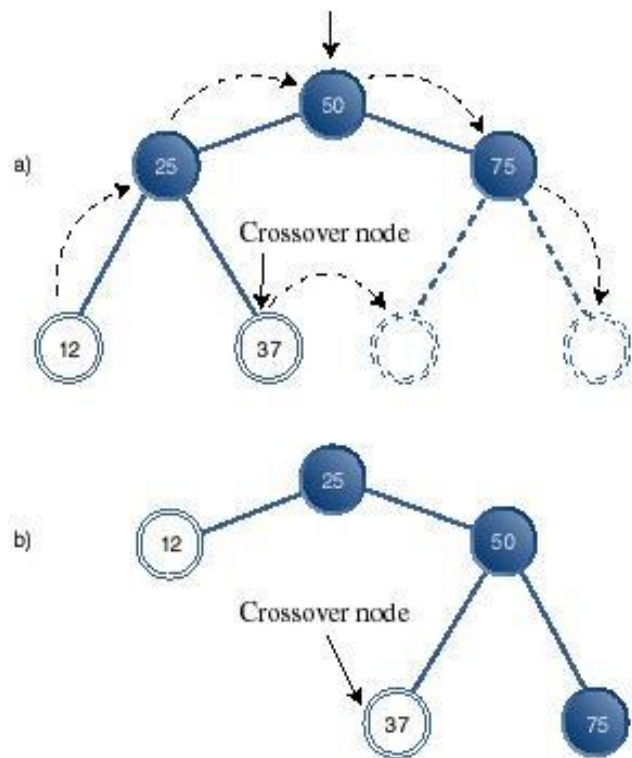
Literalmente, esto es lo que debemos hacer...

- Para una rotación a la derecha, el nodo superior debe tener un hijo izquierdo
- Entonces debemos tratar el llamado ‘nodo de cruce’, el cual debe ser movido al lado opuesto.
 - En este caso, (37)



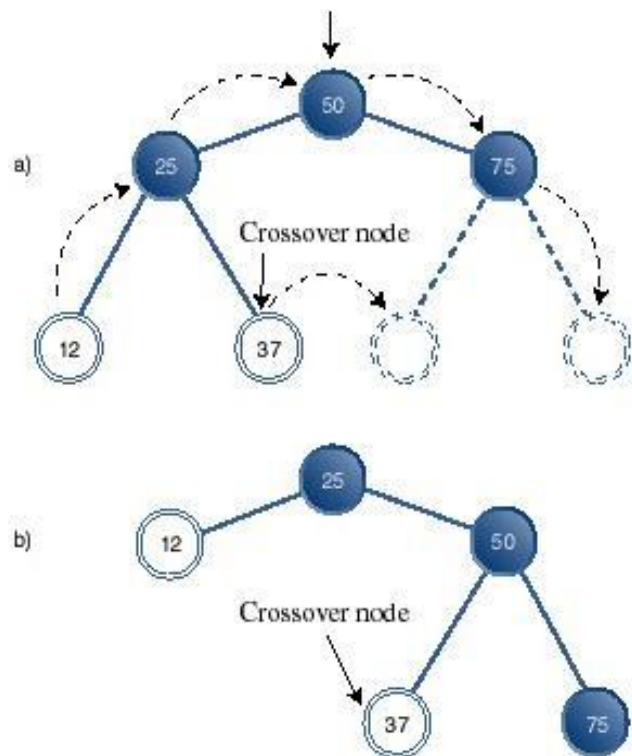
¿Quién es este bicho raro?

- Una rotación implica un nodo superior, en este caso (50)
- El nodo de cruce (37) es el hijo derecho del hijo izquierdo de (50)
 - Para una rotación a la derecha, este siempre será el caso
 - También es conocido como el nieto del interior
 - El resto se mueve a la derecha, (37) se cruza
 - SIEMPRE se reconecta con el nodo superior (50)



Así que los pasos de una rotación a la derecha son...

- Rote cada nodo, a excepción del nodo de cruce, a la derecha.
- Después de que esto sucede, el nodo superior tendrá un hijo izquierdo nulo
 - Mueva el nodo de cruce allí
- Para ganar más practica tratemos una rotación a la izquierda.



Rotación con Subárboles

- El mismo principio. Si rotamos a la derecha, con (50) como el nodo superior, podemos hacer lo mismo pero con subárboles
- (25) se convierte en la raíz
- El subárbol con (37) es el 'árbol de cruce', y se convierte en el hijo izquierdo de (50)
- Rotación con sub-árboles

