

AVL

1962 - Adelson-Velskii y Landis
árbol de búsqueda binario auto-balanceable



ABB



AVL

Problema con los ABBs

Data Structure	Time Complexity								Space Complexity
	Average				Worst				Worst
	Access	Search	Insertion	Deletion	Access	Search	Insertion	Deletion	
<u>Binary Search Tree</u>	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
<u>AVL Tree</u>	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(n)$

- Ver ejemplo: <https://www.cs.usfca.edu/~galles/visualization/BST.html>
- AVL: <https://www.cs.usfca.edu/~galles/visualization/AVLtree.html>
- <https://www.bigocheatsheet.com/>

Definición

Un AVL es un ABB donde para cada nodo del árbol se cumple que la altura del subárbol izquierdo y la altura del subárbol derecho difieren en como máximo 1.

- 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
 - $|H(T_i) - H(T_d)| \leq 1$

(ver ejemplos)

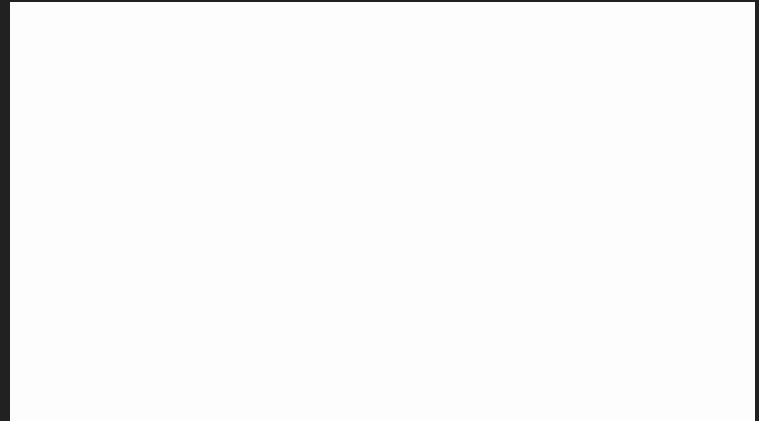
Desafío con los AVL: las rotaciones

Las rotaciones con las soluciones que hay que aplicar una vez que se nos “desequilibra” el AVL.

Momentos donde se puede generar desequilibrio ? => inserción y eliminación de elementos

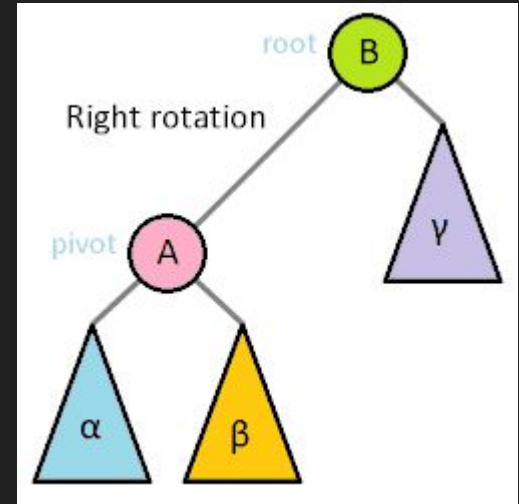
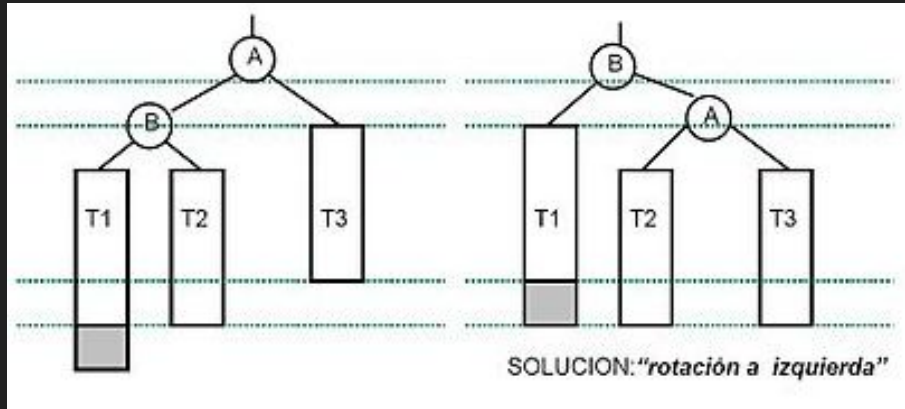
Existen 4 tipos de rotaciones.

(ver ejemplos de equilibrio => desequilibrio)



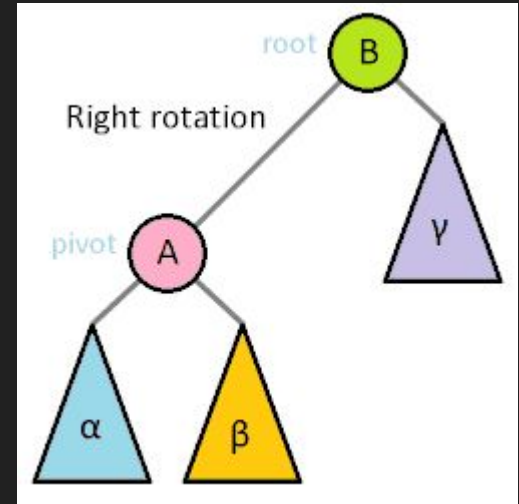
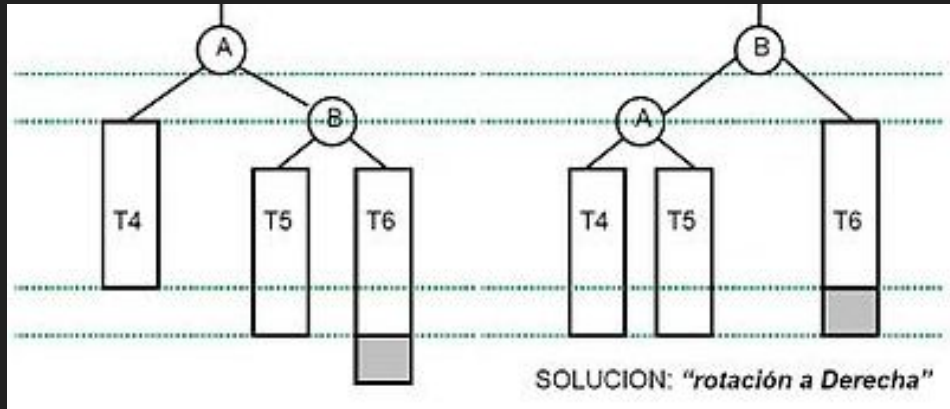
Rotación derecha simple

Ejemplo: 5 - 3 - 1



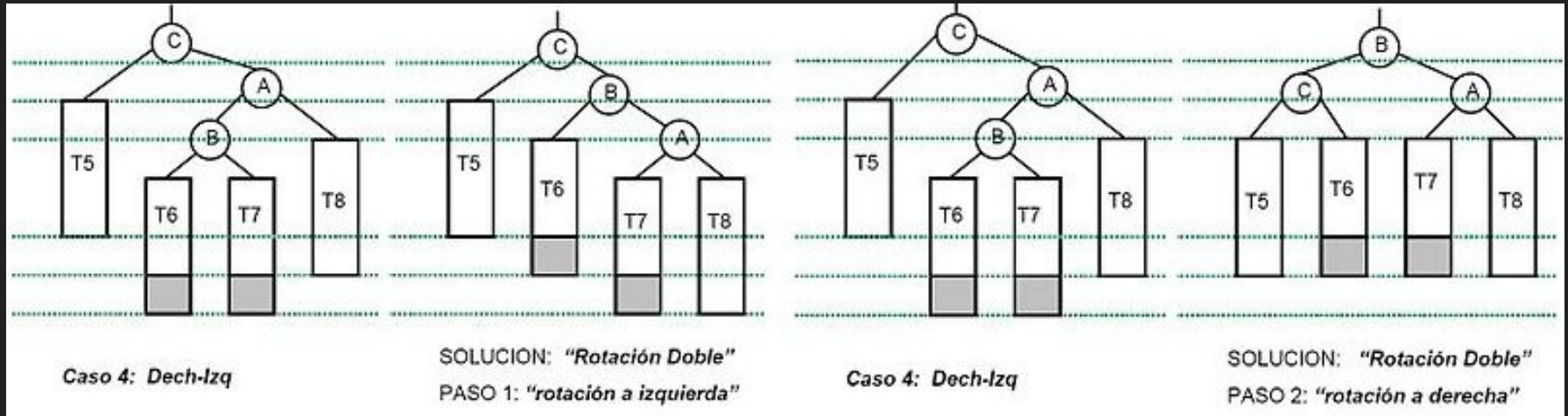
Rotación izquierda simple

Ejemplo: 1 - 3 - 5



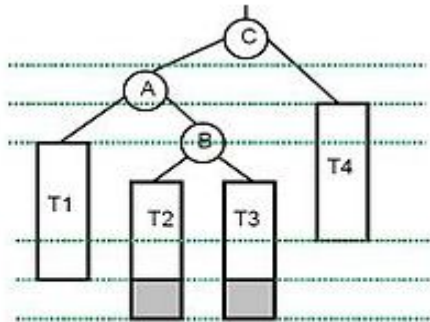
Rotación derecha doble - der-izq

Ejemplo: 10 - 5 - 15 - 4 - 6 - 20 - 16

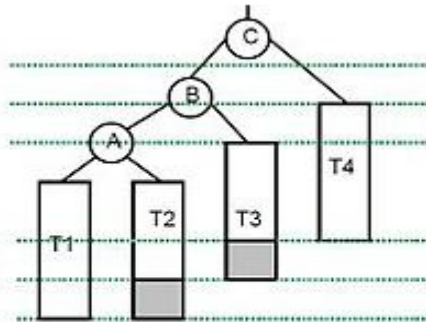


Rotación izquierda doble ó izq-der

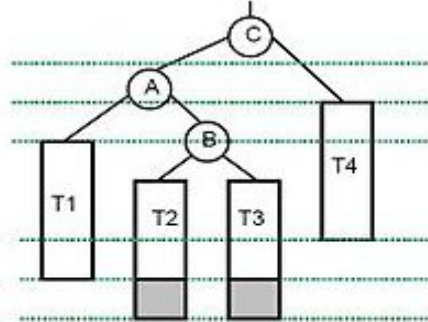
Ejemplo: 10 - 20 - 5 - 12 - 22 - 3 - 4



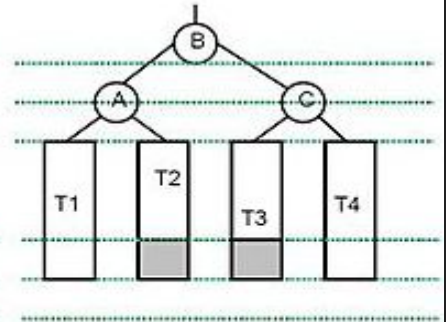
Caso 3: Izq-Dech



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



Caso 3: Izq-Dech



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



Pasos para la inserción

1) Insertar W como si fuese un ABB.

2) Empezando desde W y regresando por el camino de nodos, me encuentro (o no) con el primer nodo desbalanceado, llamémosle Z. También tenemos el nodo Y que es el hijo de Z por donde se encuentra W, y por ultimo esta el nodo X que es el nieto de Z en la ruta de Z->W.

3) Re-balancear el árbol según la rotación que le corresponda:

a) Y es el hijo izquierdo de Z y X es el hijo izquierdo de Y => rotación simple derecha

b) Y es el hijo izquierdo de Z y X es el hijo derecho de Y => rotación izquierda-derecha

c) Y es el hijo derecho de Z y X es el hijo derecho de Y => rotación simple izquierdo

d) Y es el hijo derecho de Z y X es el hijo izquierdo de Y => rotación derecha-izquierda

Nota: en los casos de rotación doble, no se aplican sobre el mismo nodo

Encuesta

<https://forms.office.com/Pages/ResponsePage.aspx?id=zSCX18DYDE2kBC3NAI8B4zHaHUx7m7VBrn630x8c1m5UQ0FYT1JSU0JGRVZIMzBHRUZIVzIXWVFDTC4u>

Links de interés

- <https://www.geeksforgeeks.org/avl-tree-set-1-insertion/>
- <https://www.geeksforgeeks.org/avl-with-duplicate-keys/>
- https://es.wikipedia.org/wiki/%C3%81rbol_AVL
- Cap 4.4, Estructuras de datos y algoritmos, Mark Allen Weiss