



AVL



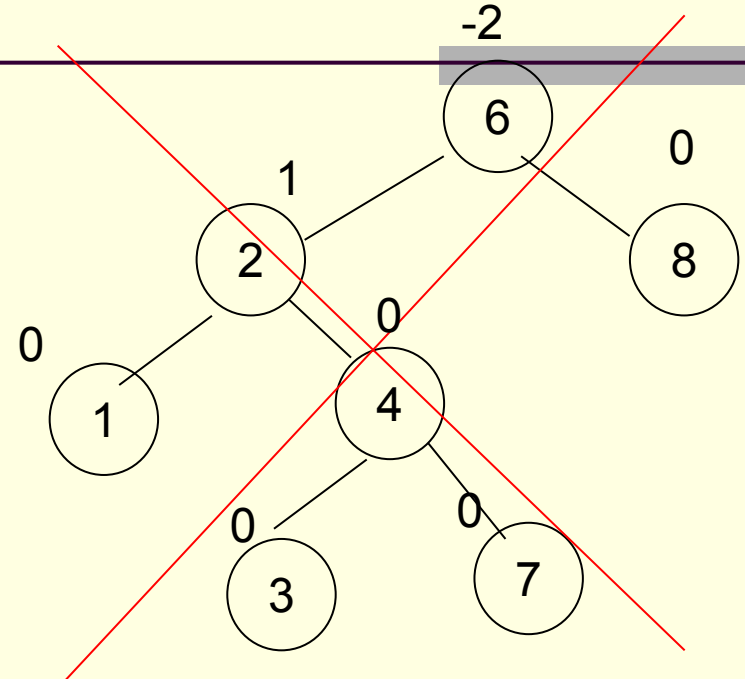
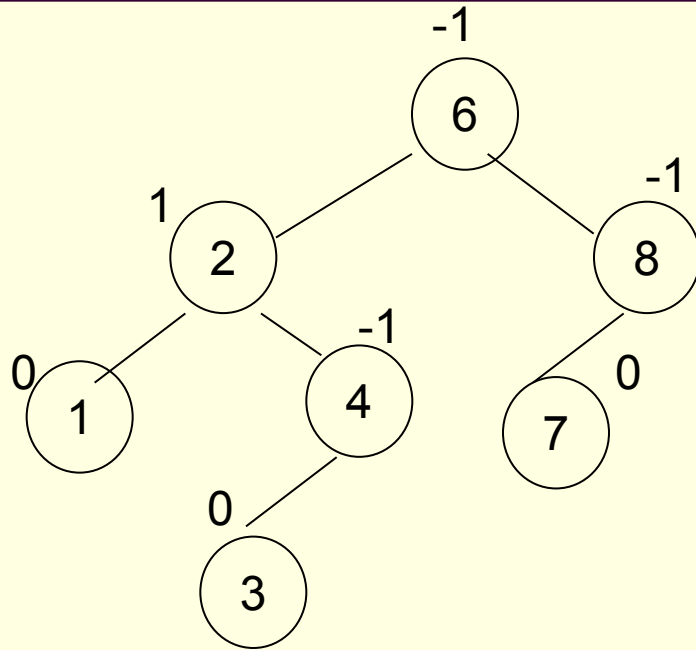
Definición

- El nombre AVL son las iniciales de los hombres que idearon este tipo de árbol **A**delson-**V**elskii y **L**andis en 1962.
- Básicamente un árbol AVL es un Árbol binario de búsqueda al que se le añade una condición de equilibrio. Esta condición es que para todo nodo la altura de sus subárboles izquierdo y derecho pueden diferir a lo sumo en 1.

Características

- Un AVL es un ABB
- La diferencia entre las alturas de los subárboles derecho e izquierdo no debe excederse en más de 1.
- Cada nodo tiene asignado un peso de acuerdo a las alturas de sus subárboles
- Un nodo tiene un peso de 1 si su subárbol derecho es más alto, -1 si su subárbol izquierdo es más alto y 0 si las alturas son las mismas.
- La inserción y eliminación en AVLs es la misma que en los ABBs.

Ejemplo de AVL



Sólo el árbol de la izquierda es AVL. El de la derecha viola la condición de equilibrio en el nodo 6, ya que su subárbol izquierdo tiene altura 3 y su subárbol derecho tiene altura 1.

Equilibrio

- $\text{Equilibrio}(n) = \text{altura_der}(n) - \text{altura_izq}(n)$
describe relatividad entre subárbol der y subárbol izq.
 - + (positivo) □ der mas alto (profundo)
 - - (negativo) □ izq mas alto (profundo)
- **Un árbol binario es un AVL si y sólo si cada uno de sus nodos tiene un equilibrio de -1, 0, + 1**
- Si alguno de los pesos de los nodos se modifica en un valor no válido (2 o -2) debe seguirse un esquema de rotación.

Operaciones sobre un AVL

- Insertar
- Balancear
 - Caso 1 Rotación simple izquierda RSI
 - Caso 2 Rotación simple derecha RSD
 - Caso 3 Rotación doble izquierda RDI
 - Caso 4 Rotación doble derecha RDD
- Eliminar
- Calcular Altura

Implementación: Clase Nodo

```
Class Nodo {  
    int bal; //para almacenar el valor del equilibrio del nodo  
    int dato;  
    Nodo izq;  
    Nodo der;  
}
```

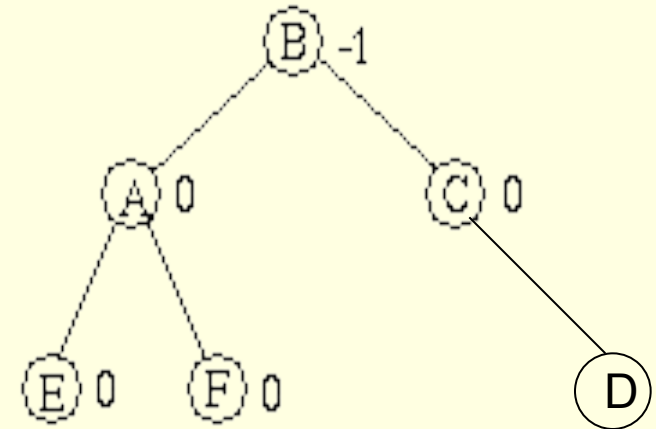
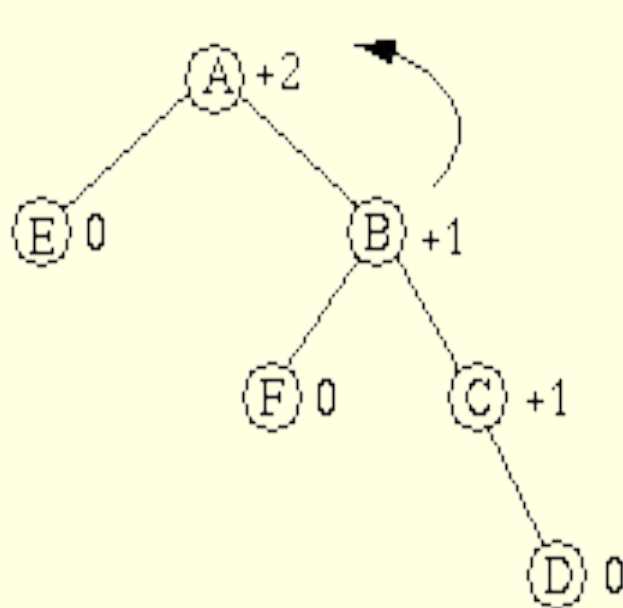
Insertar

Usamos la misma técnica para insertar un nodo en un ABB ordenado

- trazamos una ruta desde el nodo raíz hasta un nodo hoja (donde hacemos la inserción).
- Insertamos el nodo nuevo.
- Volvemos a trazar la ruta de regreso al nodo raíz, ajustando el equilibrio a lo largo de ella.
- Si el equilibrio de un nodo llega a ser $+ - 2$, volvemos a ajustar los subárboles de los nodos para que su equilibrio se mantenga acorde con los lineamientos AVL (que son $+ - 1$)

Balancear

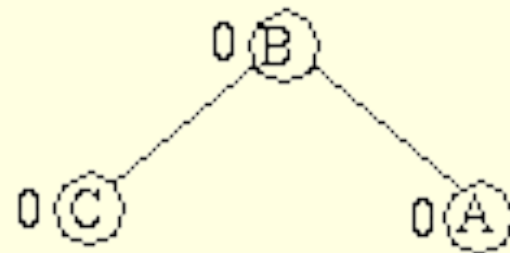
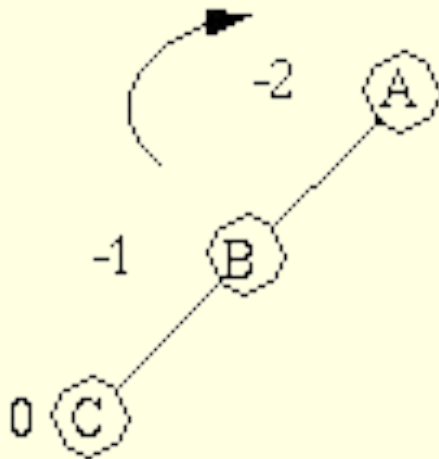
- Caso 1: Rotación simple izquierda RSI
 - Si esta desequilibrado a la izquierda y su hijo derecho tiene el mismo signo (+) hacemos rotación sencilla izquierda.



Balancear

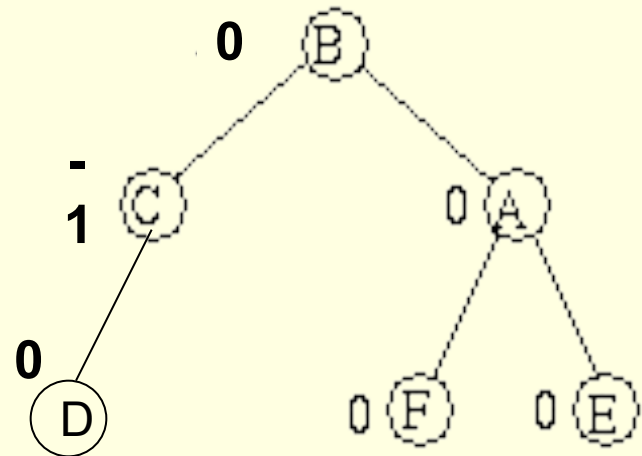
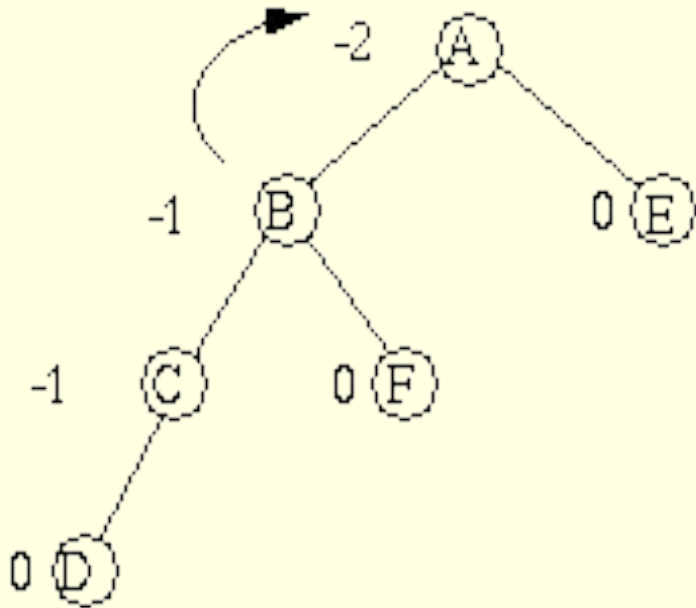
- Caso 2: Rotación simple derecha RSD
 - Si esta desequilibrado a la derecha y su hijo izquierdo tiene el mismo signo (-) hacemos rotación sencilla derecha.

Ejemplo 1



Balancear

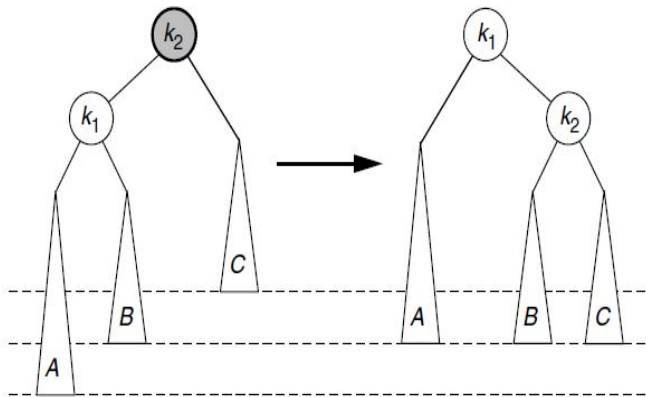
- Caso 2: Rotación simple derecha RSD
Ejemplo 2



Rotación simple izquierda o derecha.

- Hay varios puntos que cabe señalar aquí:
 - Se conserva el orden apropiado del árbol.
 - Restablece todos los nodo a equilibrios apropiados AVL
 - Conserva el recorrido en orden que el árbol anterior.
 - Sólo necesitamos modificar 3 apuntadores para lograr el nuevo equilibrio (con la de la raíz)

Código para la rotación simple (caso 1)



(a) Before rotation

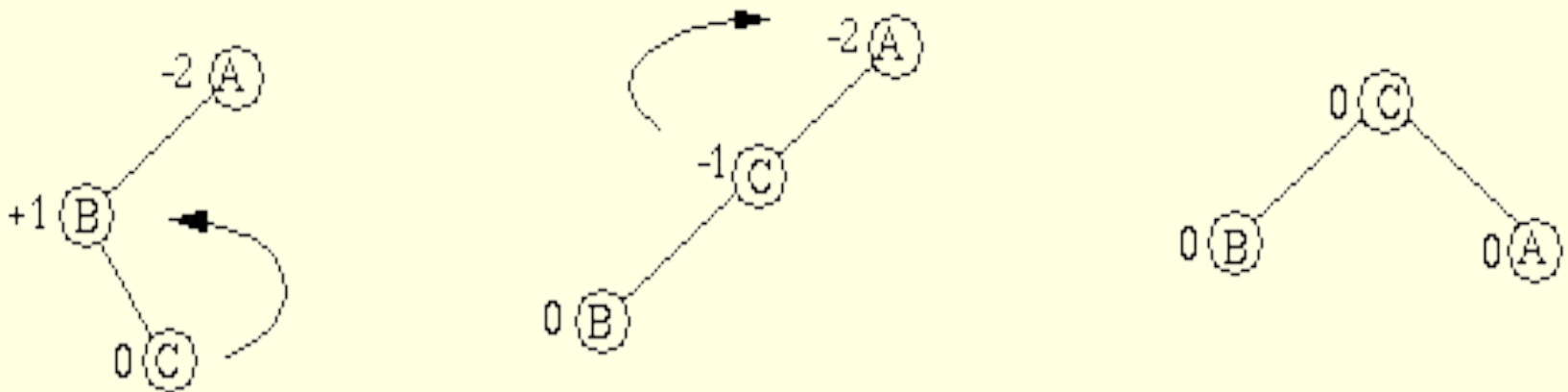
(b) After rotation

```
1  /**
2   * Rotate binary tree node with left child.
3   * For AVL trees, this is a single rotation for case 1.
4   */
5  static BinaryNode rotateWithLeftChild( BinaryNode k2 )
6  {
7      BinaryNode k1 = k2.left;
8      k2.left = k1.right;
9      k1.right = k2;
10     return k1;
11 }
```

Balancear

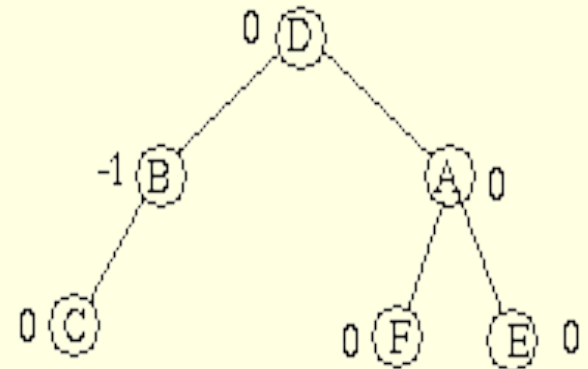
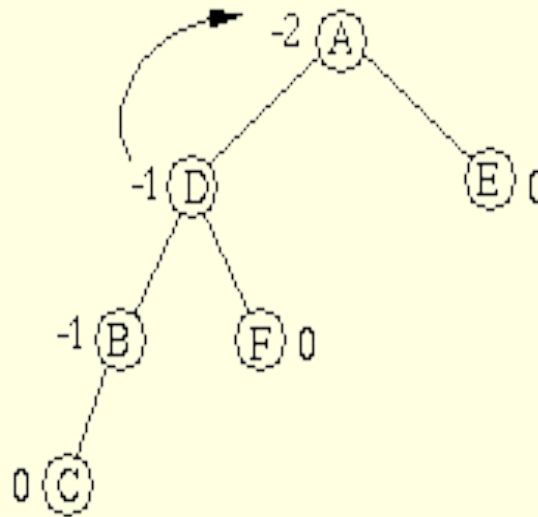
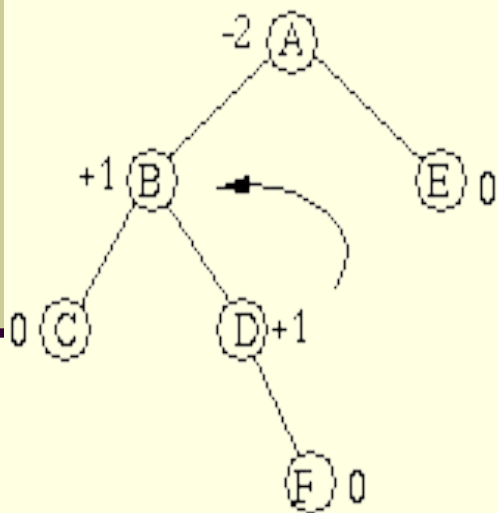
- Caso 3: Rotación doble izquierda RDI
 - Si está desequilibrado a la derecha ($FE < -1$), y su hijo izquierdo tiene distinto signo (+) hacemos rotación doble izquierda-derecha.

Ejemplo 1



Balancear

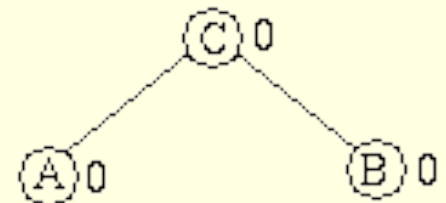
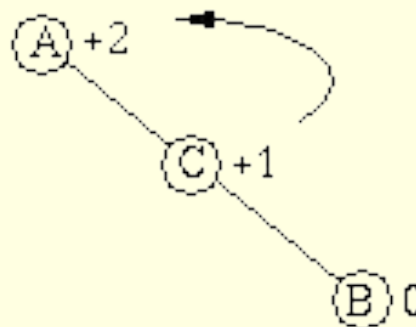
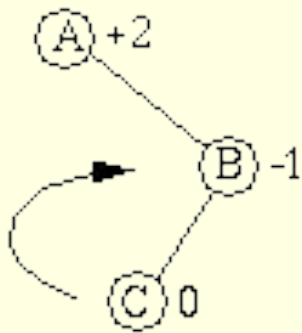
- Caso 3: Rotación doble izquierda RDI
Ejemplo 2



Balancear

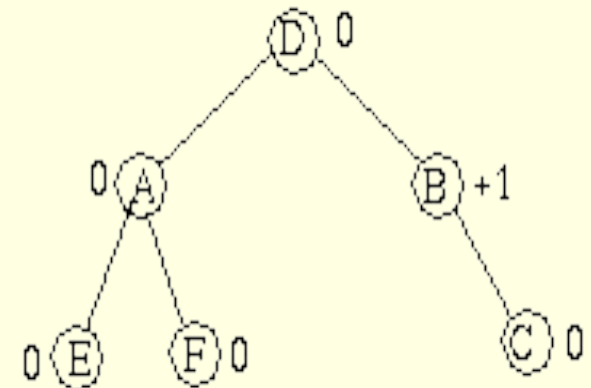
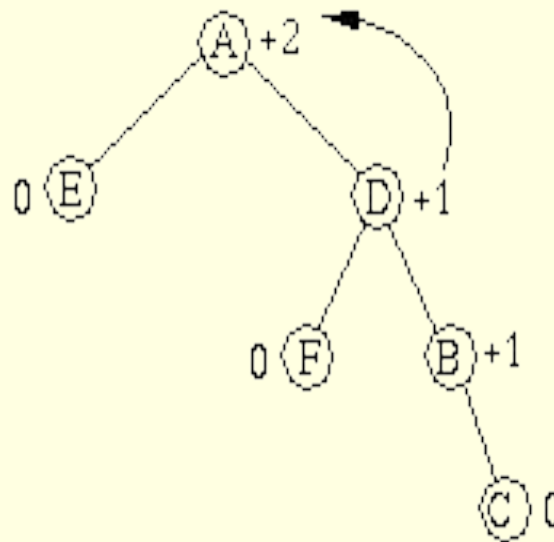
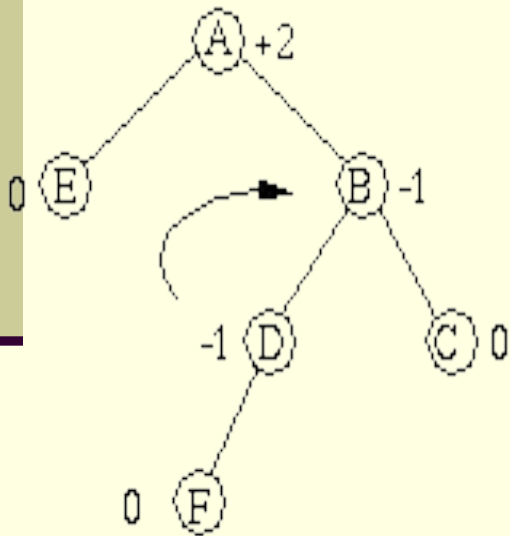
- Caso 4: Rotación doble derecha RDD
 - Si esta desequilibrado a la izquierda y su hijo derecho tiene distinto signo (−) hacemos rotación doble derecha-izquierda.

Ejemplo 1



Balancear

- Caso 4: Rotación doble derecha RDD
Ejemplo 2



Eliminar

Al eliminar un nodo en un árbol AVL puede afectar el equilibrio de sus nodos. Entonces hay que hacer rotaciones simples o dobles.

Eliminas un nodo como lo hacemos en un árbol binario ordenado. Al localizar el nodo que queremos eliminar seguimos este procedimiento:

- Si el nodo es un nodo hoja, simplemente lo eliminamos.
- Si el nodo solo tiene un hijo, lo sustituimos con su hijo.
- Si el nodo eliminado tiene dos hijos, lo sustituimos por el hijo derecho y colocamos el hijo izquierdo en el subárbol izquierdo del hijo derecho.

Ahora que hemos eliminado el nodo, tenemos que volver a equilibrar el árbol:

- Si el equilibrio del padre del nodo eliminado cambia de 0 a ± 1 el algoritmo concluye.
- Si el padre del nodo eliminado cambia de ± 1 a 0, la altura del árbol ha cambiado y se afecta el equilibrio de su abuelo.
- Si el equilibrio del padre del nodo eliminado cambia de ± 1 a ± 2 hay que hacer una rotación. Después de concluirla, el equilibrio del padre podría cambiar, lo que, a su vez, podría forzarnos a hacer otros cambios (y probables rotaciones) en toda la ruta hacia arriba a medida que ascendemos hacia la raíz. Si encontramos en la ruta un nodo que cambie de 0 a ± 1 entonces terminamos.