

Ejercicio: Genere el árbol binario de búsqueda para la siguiente secuencia de números: 8, 9, 11, 15, 19, 20, 21, 7, 3, 2, 1, 5, 6, 4, 13, 14, 10, 12, 17, 16, 18. Analice y describa lo que sucede durante su inserción.

Si de alguna manera se pudiera garantizar que el árbol además de que no es degenerado mantiene un balance perfecto, entonces el mecanismo de eficiencia de búsqueda en un árbol binario se mantendría inalterado.

La idea es aquí entonces, generar un árbol de altura mínima que contenga el mismo número de nodos.

8.4.1 Árboles perfectamente balanceados.

Se dice que un árbol es **perfectamente balanceado**, si para cada nodo en el árbol, el número de nodos en sus subárboles izquierdo y derecho difieren a lo más en 1.

Ejercicio: Dado un conjunto de elementos a insertar dentro de un árbol binario perfectamente balanceado ¿En qué orden deben insertarse los nodos para que el árbol permanezca inalterado respecto a su balance?

Considere el siguiente algoritmo para generar un árbol binario perfectamente balanceado:

1. Sea ***n*** el número de nodos a insertar
2. Utilizar un nodo para la raíz
3. Generar el subárbol izquierdo recursivamente con el siguiente número de nodos:

$$\mathbf{n_{izq} = n / 2}$$

4. Generar el subárbol derecho recursivamente con el siguiente número de nodos:

$$\mathbf{n_{der} = n - n_{izq} - 1}$$

Ejercicio: En base al algoritmo propuesto, genere el árbol binario perfectamente balanceado para la siguiente secuencia de números: 8, 9, 11, 15, 19, 20, 21, 7, 3, 2, 1, 5, 6, 4, 13, 14, 10, 12, 17, 16, 18.

El algoritmo anterior asume que de antemano se conoce el número total de elementos a insertar, lo cual no siempre es posible, por otro lado, debe observarse como resultado del ejercicio que el árbol no mantiene un orden respecto al los elementos que contiene.

Mantener un árbol perfectamente balanceado es una operación muy costosa, por lo que conviene buscar un criterio alternativo que proporcione ventajas similares pero a un menor costo.

8.4.2 Árboles balanceados.

Un **árbol binario balanceado** (**árbol AVL – Adelson-Velski y Landis**) es aquel en el que las alturas de los dos subárboles de cada nodo difieren a lo más en 1.

El **balance** de un nodo en un árbol binario en general, y de un árbol AVL en particular, puede definirse como la altura de su subárbol izquierdo menos la altura de su subárbol derecho. Por conveniencia, la altura de un árbol nulo se define como -1.

Existen básicamente dos casos que corrigen el rebalanceo de un árbol AVL, estos casos ilustran el proceso general de rebalanceo que se aplica.

Se mencionan solamente dos casos debido a que los otros casos son simétricos y puede derivarse fácilmente de los que se presentan.

El primero de ellos muestra el proceso de rotación simple y se presenta en la Ilustración 6.

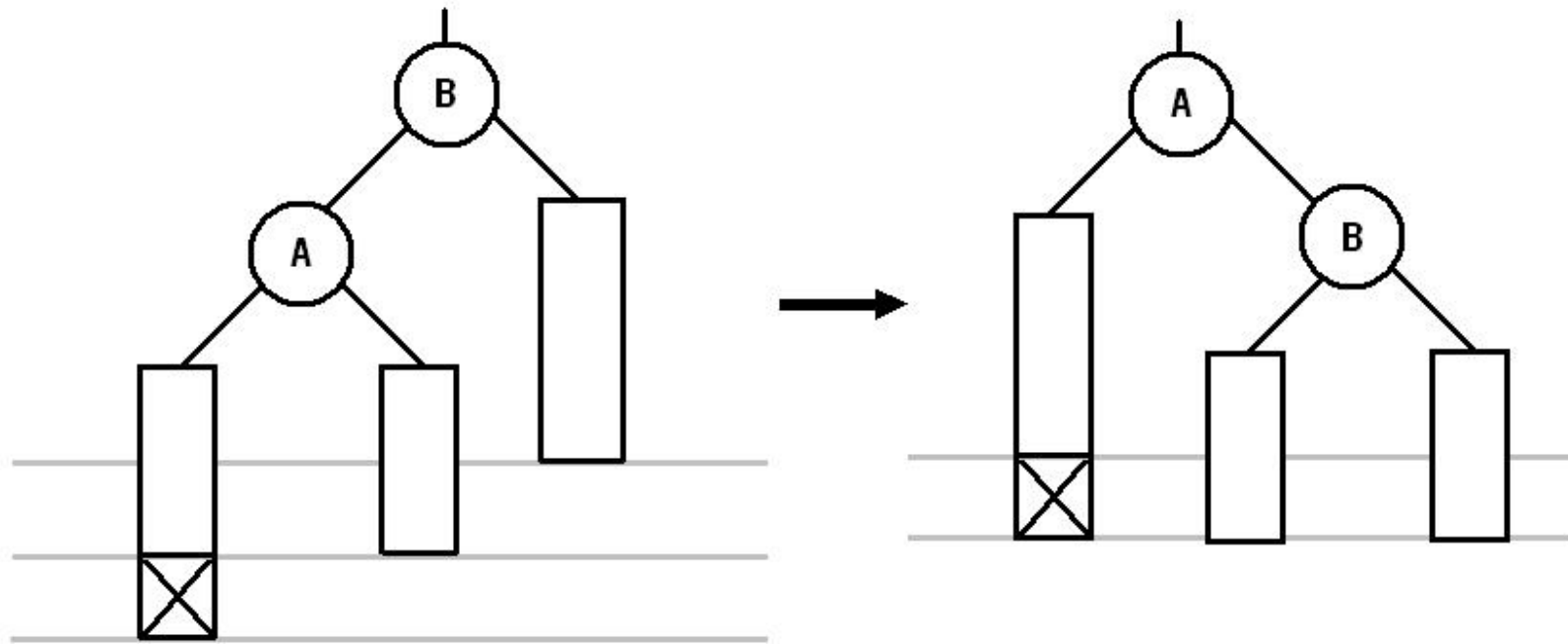


Ilustración 6. Caso 1 de rebalanceo: rotación sencilla.

El segundo de ellos es un proceso un poco más elaborado ya que implica dos rotaciones, las cuales se muestran en la Ilustración 7.

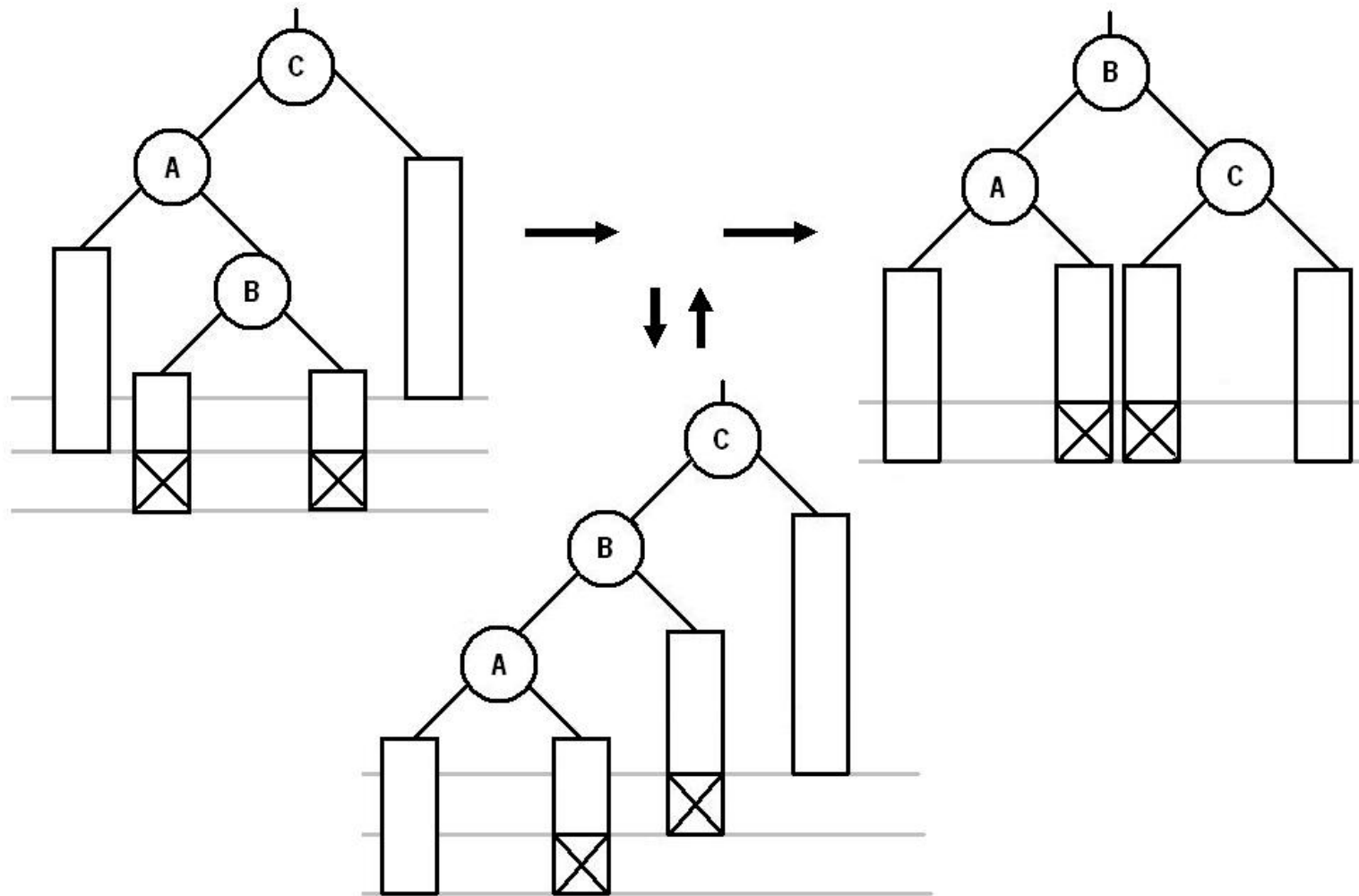


Ilustración 7. Caso 2 de rebalanceo: rotación doble.

Aunque los diagramas anteriores deberían ser suficientes para entender el proceso de rebalanceo, se utilizará un sencillo ejemplo que los ilustre.

Considérese el árbol AVL que aparece en el inciso **(a)** de la Ilustración 8. El caso 1 se presenta al insertar los nodos **1** o **3**, mismos que se muestran en la parte **(b)**. La aplicación del caso 1 dará como resultado el árbol AVL que aparece en **(c)** dependiendo del nodo que se haya insertado.

Asegúrese de que entiende el proceso de rebalanceo aplicado.

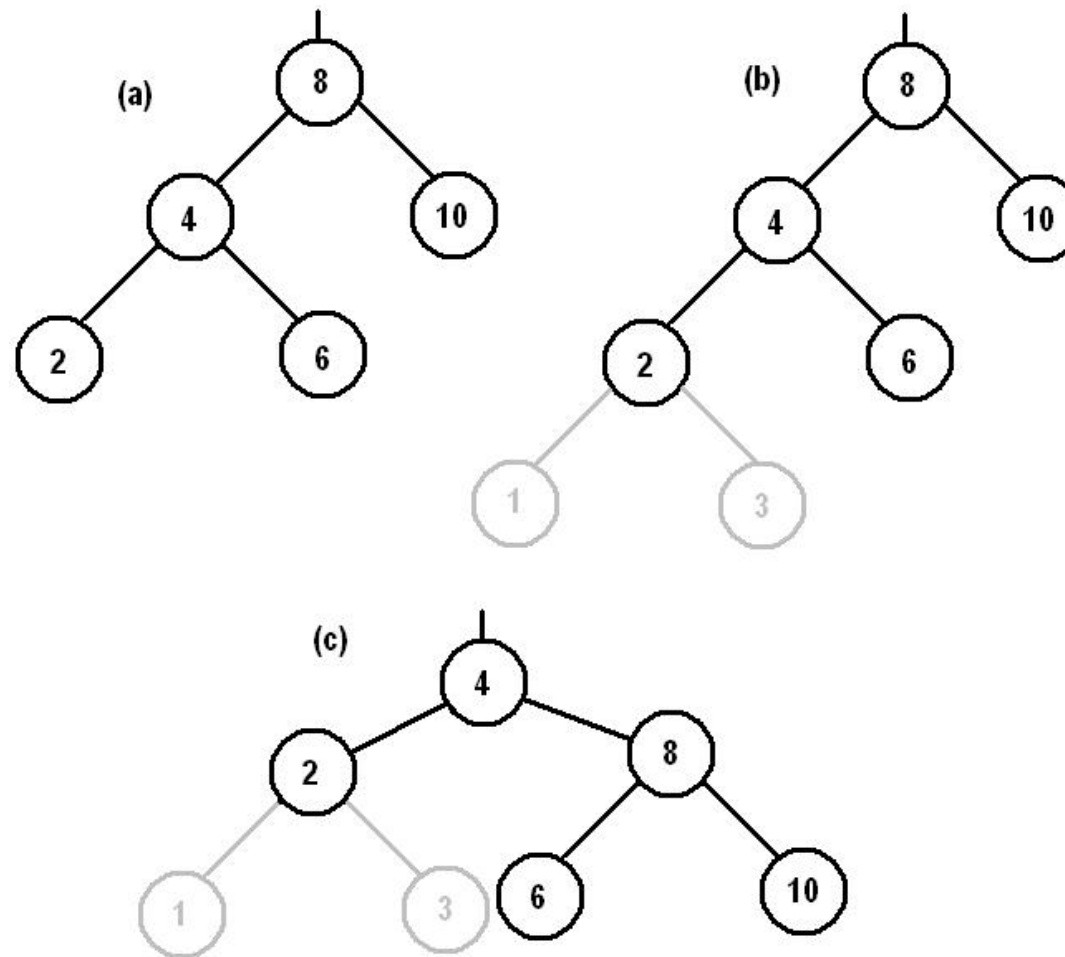


Ilustración 8. Ejemplo de la aplicación del caso 1 de rebalanceo.

Para el segundo caso, considérese nuevamente el mismo árbol AVL que aparece en el inciso **(a)** de la Ilustración 9. Ahora el caso 2 se presenta al insertar los nodos **5** o **7**, mismos que se muestran en la parte **(b)**. La aplicación del caso 2 dará como resultado el árbol AVL que aparece en **(c)** dependiendo del nodo que se haya insertado.

Antes de continuar, asegúrese de que entiende los procesos de rebalanceo aplicados.

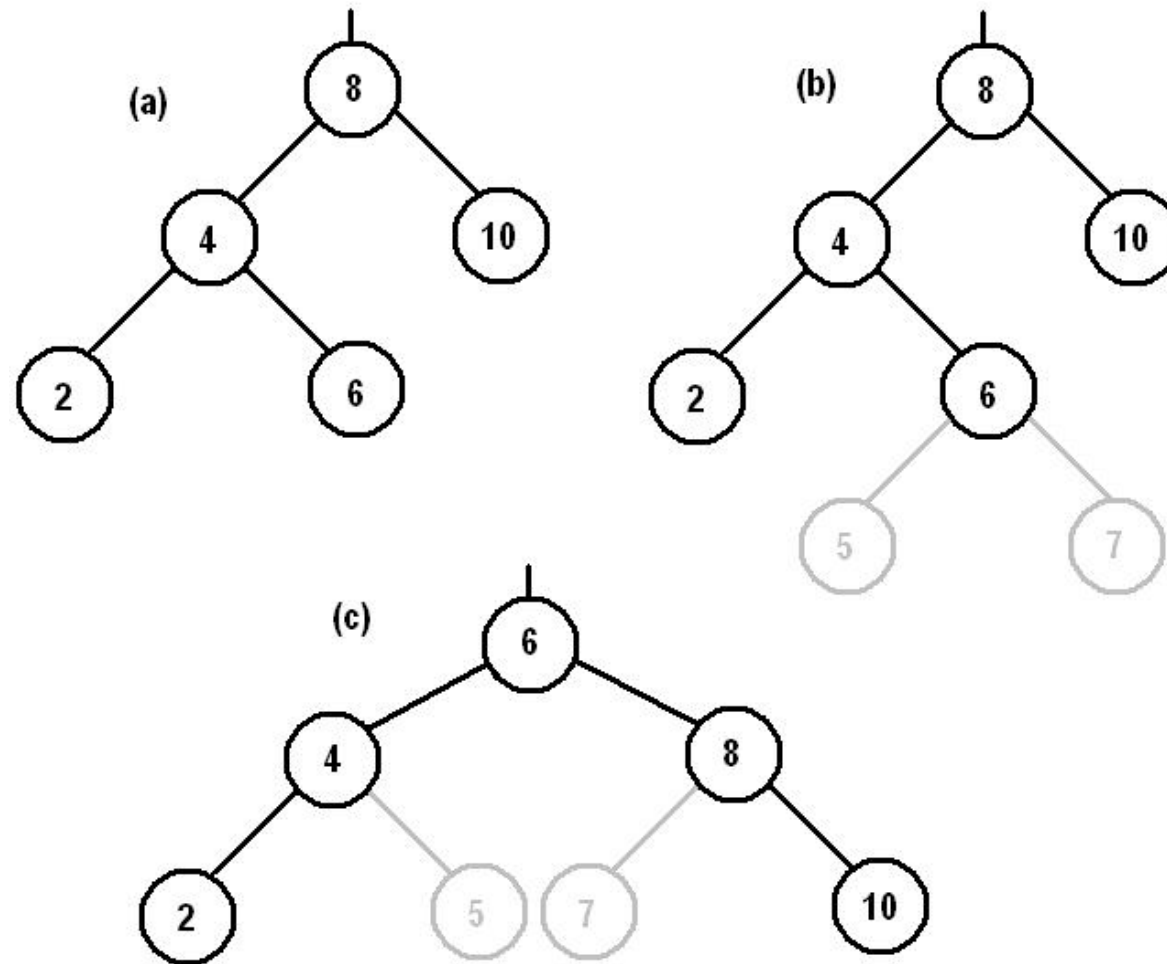


Ilustración 9. Ejemplo de la aplicación del caso 2 de rebalanceo.

Ejercicio 1: Dada la siguiente secuencia de números: **4, 5, 7, 2, 1, 3, 6**, generar un árbol AVL. Se deberá ilustrar paso a paso el proceso de inserción y rebalanceo.

Ejercicio 2: Dada la siguiente secuencia de números: **8, 9, 11, 15, 19, 20, 21, 7, 3, 2, 1, 5, 6, 4, 13, 14, 10, 12, 17, 16, 18**, generar:

- a) Su árbol binario de búsqueda
- b) Su árbol perfectamente balanceado
- c) Su árbol AVL

Se deberá ilustrar paso a paso en cada uno de ellos, el proceso de inserción.

Ejercicio 3: ¿Qué conjeturas puede sacar de estos ejercicios? ¿En qué tipo de árbol se genera el árbol de altura mínima? ¿Cuáles serían las ventajas y desventajas de uno y otro esquema de representación de árbol?

Todo lo hasta ahora visto respecto a árboles AVL está relacionado con el rebalanceo después de la inserción de algún nodo que rompa el balance del árbol AVL. Sin embargo ¿qué pasa con la eliminación de elementos?