Estructuras de Datos
Árboles

Tópicos

- Búsqueda binaria
- Árboles de búsqueda binaria
  - Búsqueda en un ABB
  - Inserción en un ABB
  - Eliminación en un ABB
- Árboles AVL
  - Inserción en un AVL
    - Rotación Simple
    - Rotación Doble
  - Eliminación en un AVL

Búsqueda binaria

Suponga que se dispone del arreglo `a`, de tamaño `n`, en donde se tiene almacenado el conjunto de elementos ordenados de menor a mayor. Para buscar un elemento `x` dentro del arreglo se debe:

- buscar el índice de la posición media del arreglo en donde se busca el elemento, que se denotará `m`. Inicialmente, `m = n/2`.
- si `a[m]=x` se encontró el elemento (fin de la búsqueda), en caso contrario se sigue buscando en el lado derecho o izquierdo del arreglo dependiendo si `a[m]<x` ó `a[m]>x` respectivamente.

Búsqueda binaria

La variable `i` representa el primer casillero del arreglo en donde es posible que se encuentre el elemento x, la variable `j` representa el último casillero del arreglo hasta donde x puede pertenecer, con `j>=i`.

Inicialmente, `i=0` y `j=n-1`.

Búsqueda binaria

En cada iteración,

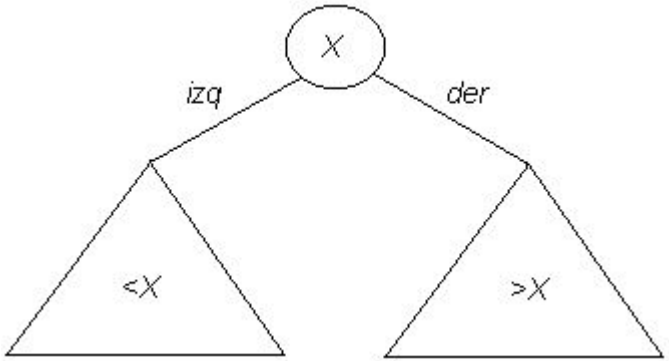
- Si el conjunto es vacío (`j-i < 0`), o sea si `j < i`, entonces el elemento `x` no está en el conjunto (búsqueda infructuosa).
- En caso contrario, `m = (i+j)/2`.
  Si `x=a[m]`, el elemento fue encontrado (búsqueda exitosa).
  Si `x<a[m]` se modifica `j=m-1`, sino se modifica `i=m+1` y se sigue iterando.

Búsqueda binaria

```
#define NO_ENCONTRADO -1;
int busquedaBinaria(int* a, int n, int x) {
  int i=0, j=n-1, m;
  while (i<=j) {
    m=(i+j)/2;
    if (x==a[m]) {return m;}
    if (x<a[m]) {
      j=m-1;
    }
    else {
      i=m+1;
    }
  }
  return NO_ENCONTRADO;
}
```

Árboles de búsqueda binaria

Un Árbol de búsqueda binaria, en adelante ABB, es un árbol binario en donde todos los nodos cumplen la siguiente propiedad (sin perder generalidad se asumirá que los elementos almacenados son números enteros): si el valor del elemento almacenado en un nodo N es X, entonces todos los valores almacenados en el subárbol izquierdo de N son menores que X, y los valores almacenados en el subárbol derecho de N son mayores que X.



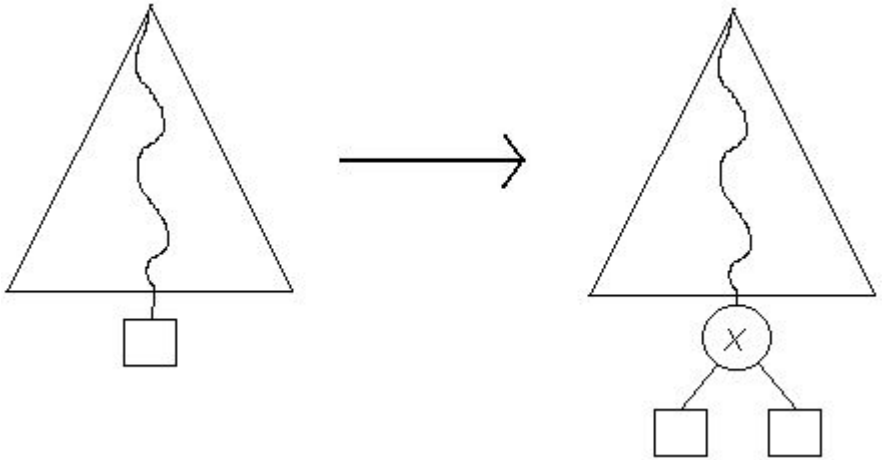
Búsqueda en un ABB

Esta operación retorna un puntero al nodo en donde se encuentra el elemento buscado, x, o NULL si dicho elemento no se encuentra en el árbol. La estructura del árbol facilita la búsqueda:

- Si el árbol está vacío, entonces el elemento no está y se retorna NULL.
- Si el árbol no está vacío y el elemento almacenado en la raíz es X, se encontró el elemento y se retorna un puntero a dicho nodo.
- Si X es menor que el elemento almacenado en la raíz se sigue buscando recursivamente en el subárbol izquierdo, y si X es mayor que el elemento almacenado en la raíz se sigue buscando recursivamente en el subárbol derecho.

Inserción en un ABB

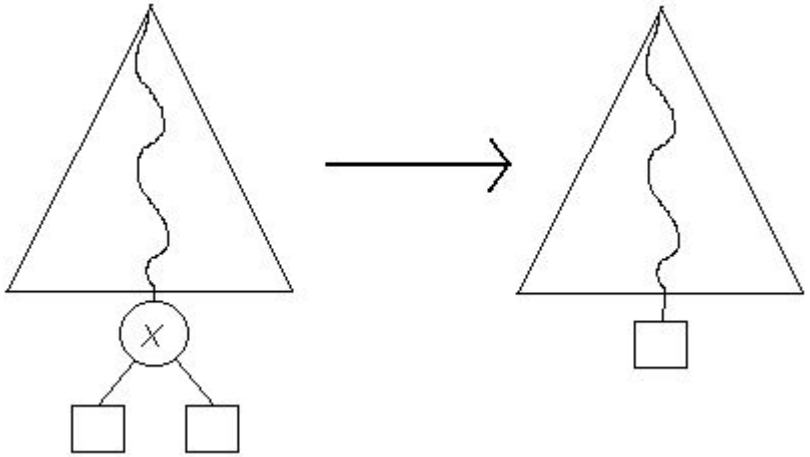
Para insertar un elemento X en un ABB, se realiza una búsqueda infructuosa de este elemento en el árbol, y en el lugar en donde debiera haberse encontrado se inserta.



Eliminación en un ABB

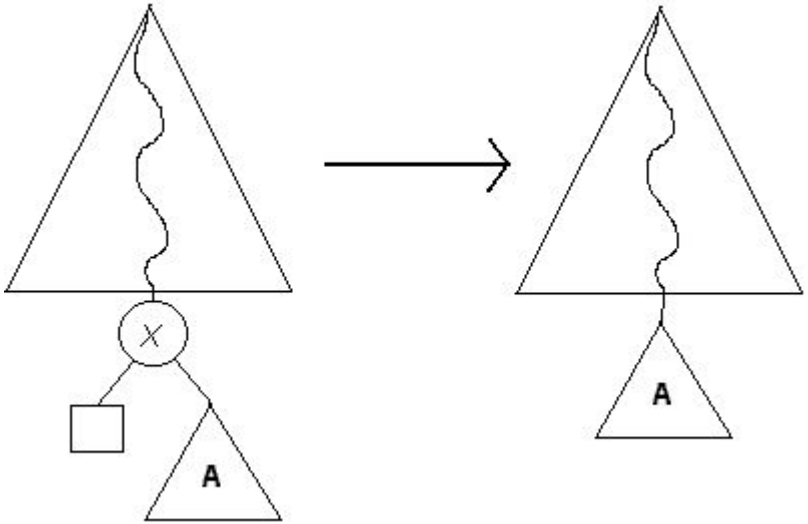
Primero se realiza una búsqueda del elemento a eliminar, digamos X. Si la búsqueda fue infructuosa no se hace nada, en caso contrario hay que considerar los siguientes casos posibles:

1. Si X es una hoja sin hijos, se puede eliminar inmediatamente.



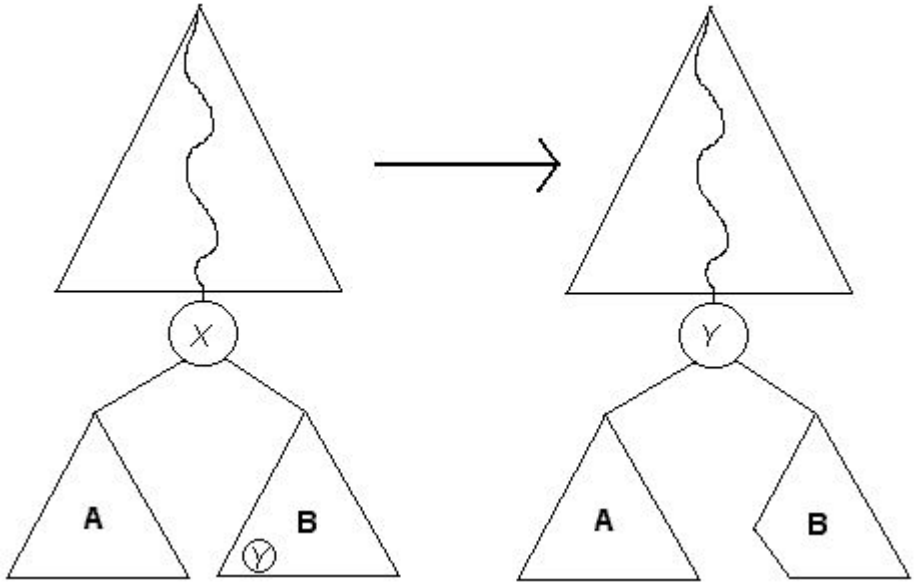
Eliminación en un ABB

2. Si X tiene un solo hijo, entonces se cambia la referencia del padre a X para que ahora referencie al hijo de X.



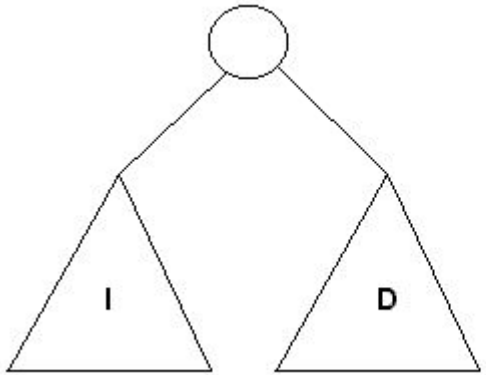
Eliminación en un ABB

3. Si X tiene dos hijos, el caso complicado, se procede de la siguiente forma: se elimina el nodo Y = mínimo nodo del subárbol derecho de X, el cual corresponde a uno de los casos fáciles de eliminación, y se reemplaza el valor almacenado en el nodo X por el valor que estaba almacenado en el nodo Y.



Árboles AVL

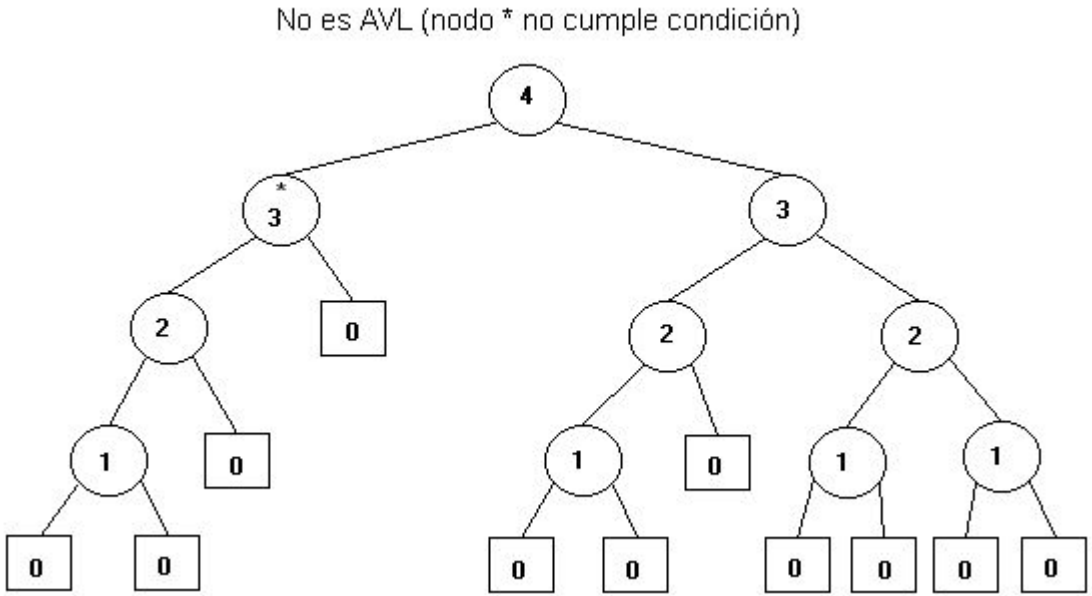
Un árbol se dice que es un Árbol AVL si para todo nodo interno la diferencia de altura de sus dos árboles hijos es menor o igual que 1.



$$|altura(I) - altura(D)| \leq 1$$

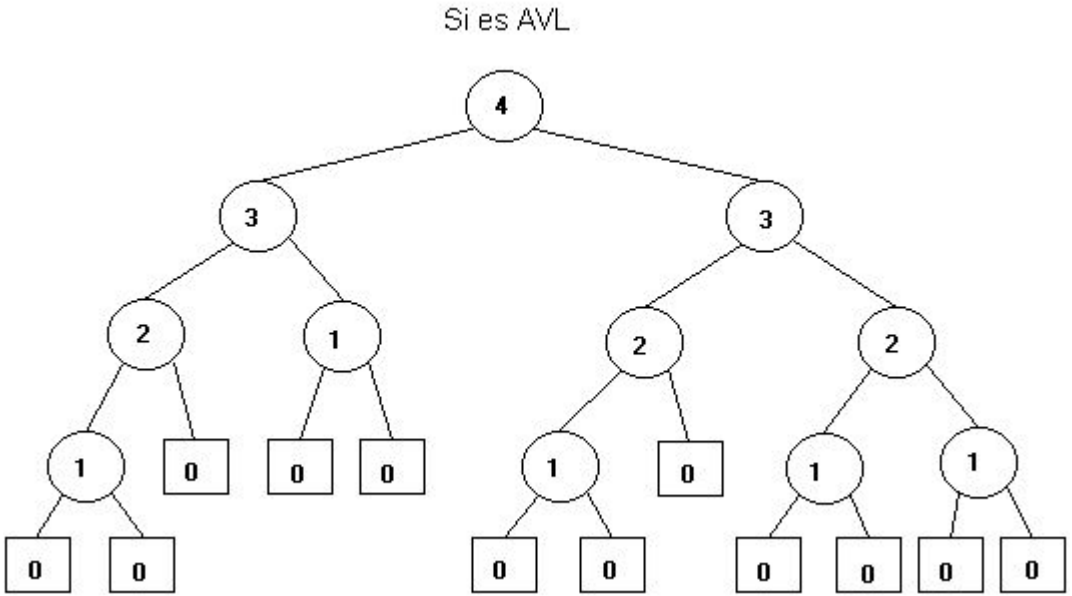
Árboles AVL

Por ejemplo: (el número dentro de cada nodo indica su altura)



Árboles AVL

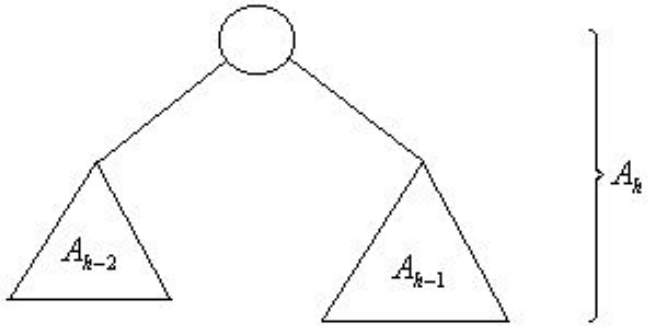
Por ejemplo: (el número dentro de cada nodo indica su altura)



Árboles AVL

Problema: para una altura h dada, ¿cuál es el árbol AVL con mínimo número de nodos que alcanza esa altura?

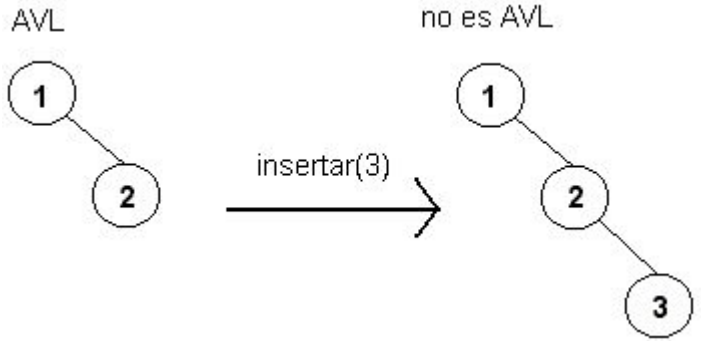
Nótese que en dicho árbol AVL la diferencia de altura de sus hijos en todos los nodos tiene que ser 1.



Inserción en un AVL

La inserción en un AVL se realiza de la misma forma que en un ABB, con la salvedad que hay que modificar la información de la altura de los nodos que se encuentran en el camino entre el nodo insertado y la raíz del árbol.

El problema potencial que se puede producir después de una inserción es que el árbol con el nuevo nodo no sea AVL:



Inserción en un AVL

En el ejemplo de la figura, la condición de balance se pierde al insertar el número 3 en el árbol, por lo que es necesario restaurar de alguna forma dicha condición. Esto siempre es posible de hacer a través de una modificación simple en el árbol, conocida como rotación.

Suponga que después de la inserción de un elemento X el nodo desbalanceado más profundo en el árbol es N. Esto quiere decir que la diferencia de altura entre los dos hijos de N tiene que ser 2, puesto que antes de la inserción el árbol estaba balanceado.

Inserción en un AVL

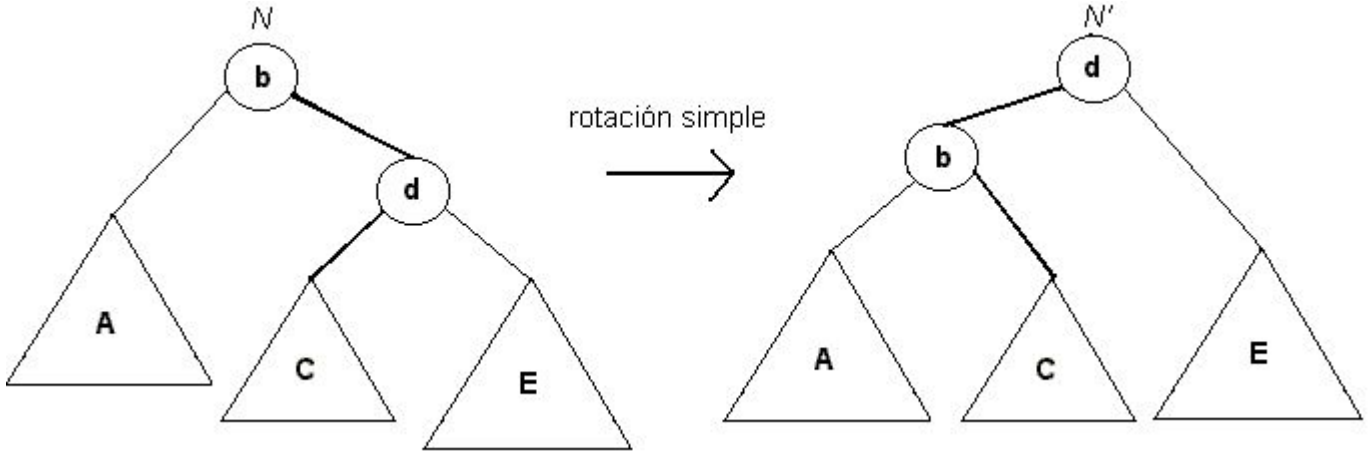
La violación del balance pudo ser ocasionada por alguno de los siguientes casos:

- El elemento X fue insertado en el subárbol izquierdo del hijo izquierdo de N.
- El elemento X fue insertado en el subárbol derecho del hijo izquierdo de N.
- El elemento X fue insertado en el subárbol izquierdo del hijo derecho de N.
- El elemento X fue insertado en el subárbol derecho del hijo derecho de N.

Dado que el primer y último caso son simétricos, así como el segundo y el tercero, sólo hay que preocuparse de dos casos principales: una inserción "hacia afuera" con respecto a N (primer y último caso) o una inserción "hacia adentro" con respecto a N (segundo y tercer caso).

Inserción en un AVL – Rotación simple

El desbalance por inserción "hacia afuera" con respecto a N se soluciona con una rotación simple.



La figura muestra la situación antes y después de la rotación simple, y ejemplifica el cuarto caso anteriormente descrito, es decir, el elemento X fue insertado en E, y b corresponde al nodo N.

Inserción en un AVL – Rotación simple

Antes de la inserción, la altura de N es la altura de C+1.

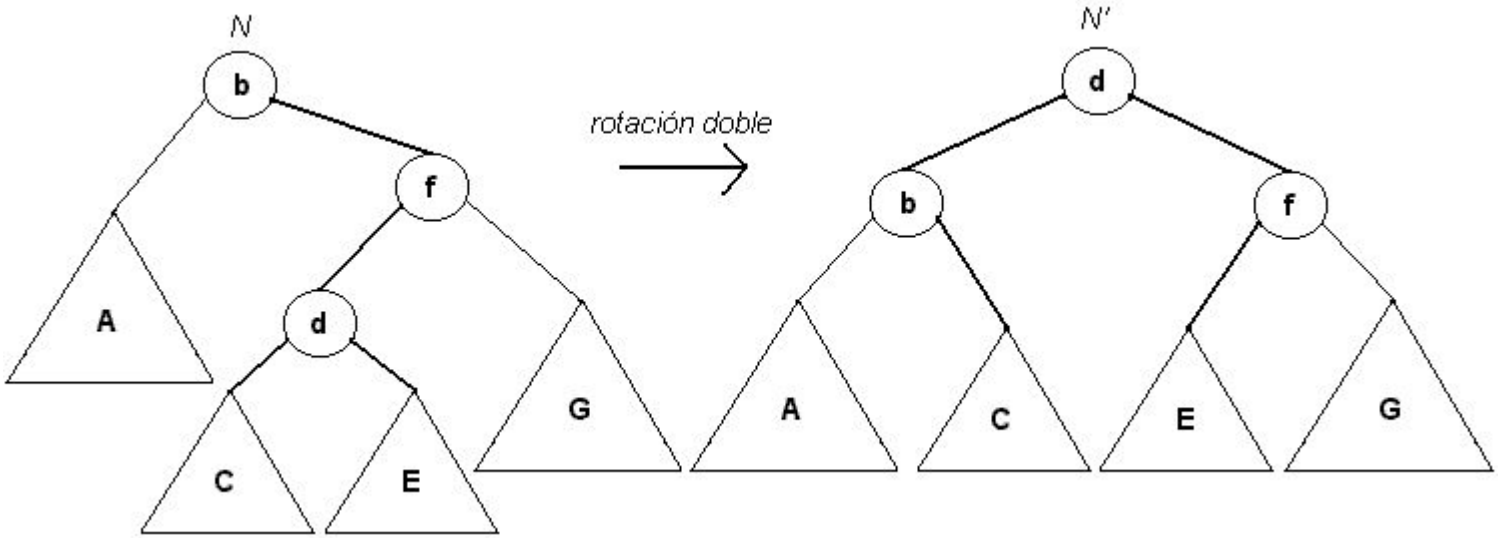
Idealmente, para recuperar la condición de balance se necesitaría bajar A en un nivel y subir E en un nivel, lo cual se logra cambiando las referencias derecha de b e izquierda de d, quedando este último como nueva raíz del árbol, N'.

Notemos que ahora el nuevo árbol tiene la misma altura que antes de insertar el elemento, C+1, lo cual implica que no puede haber nodos desbalanceados más arriba en el árbol, por lo que es necesaria una sola rotación simple para devolver la condición de balance al árbol.

El árbol sigue cumpliendo con la propiedad de ser ABB.

Inserción en un AVL – Rotación doble

Claramente un desbalance producido por una inserción "hacia adentro" con respecto a N no es solucionado con una rotación simple, dado que ahora es C quien produce el desbalance y como se vio anteriormente este subárbol mantiene su posición relativa con una rotación simple.



Inserción en un AVL – Rotación doble

Para el caso de la figura (tercer caso), la altura de N antes de la inserción era G+1. Para recuperar el balance del árbol es necesario subir C y E y bajar A, lo cual se logra realizando dos rotaciones simples: la primera entre d y f, y la segunda entre d, ya rotado, y b, obteniéndose el resultado de la figura.

A este proceso de dos rotaciones simples se le conoce como rotación doble, y como la altura del nuevo árbol N' es la misma que antes de la inserción del elemento, ningún elemento hacia arriba del árbol queda desbalanceado, por lo que solo es necesaria una rotación doble para recuperar el balance del árbol después de la inserción.

El nuevo árbol cumple con la propiedad de ser ABB después de la rotación doble.

Eliminación en un AVL

La eliminación en árbol AVL se realiza de manera análoga a un ABB, pero también es necesario verificar que la condición de balance se mantenga una vez eliminado el elemento.

En caso que dicha condición se pierda, será necesario realizar una rotación simple o doble dependiendo del caso, pero es posible que se requiera más de una rotación para reestablecer el balance del árbol.