

1) ““Los árboles AVL siempre son mejores que los ABB””

Esto es cierto en cuanto a eficiencia de los algoritmos, ya que nos permite tener el árbol continuamente balanceado por que evita que se provoque una degeneración del árbol.

Sin embargo, a los métodos tradicionales de un árbol ABB, se le suma complejidad ya que después de cada alta() y baja(), habría que chequear que todo el árbol se encuentre balanceado y balancearlo en caso de que no lo esté. Además, dependiendo de cuantos hijos tenga y para que lado este desbalanceado, los movimientos que deben ejecutarse para balancear el árbol cambiarán, lo que le agrega más dificultad para programar.

2) Método buscar(Nodo* raiz, int numero_a_buscar) de un ABB:

```
Nodo* buscar(Nodo* raiz, int numero_a_buscar){
    if(raiz == nullptr || raiz -> obtener_valor() == numero_a_buscar){
        return raiz;
    }

    Nodo* siguiente_nodo;

    if(raiz -> obtener_valor() < numero_a_buscar){
        siguiente_nodo = raiz -> obtener_hijo_derecho()
    }

    if(raiz -> obtener_valor() > numero_a_buscar){
        siguiente_nodo = raiz -> obtener_hijo_izquierdo()
    }

    return buscar(siguiente_nodo, numero_a_buscar);
}
```