

Árbol binario de búsqueda perezoso

Un árbol binario de búsqueda perezoso es un árbol binario de búsqueda que se balancea solo cuando en un nodo el subárbol izquierdo es el doble de grande que el subárbol derecho o viceversa. Cuando esto sucede se rebalancea el subárbol con la mayor cantidad de nodos.

Cuando un subárbol de un ABB perezoso es rebalanceado dicho subárbol se convierte en un arreglo ordenado. Luego ese arreglo ordenado es convertido de nuevo en un subárbol. Dicha conversión siempre resultará en un árbol balanceado debido al ordenamiento del arreglo. El punto medio se puede encontrar en tiempo constante y queda como la raíz del nuevo árbol. Al hacer esto, se insertan todos los nodos menores a la raíz para que se construya el subárbol izquierdo y después se insertan todos los nodos mayores a la raíz para que se construya el subárbol derecho. El resultado de esta operación es un ABB lo más balanceado posible.

El proyecto consiste en programar un ABB perezoso con las siguientes especificaciones:

- Para efectos de un arreglo ordenado se utilizará una lista que se comportará como una cola con prioridad con la salvedad de que tendrá una referencia adicional que siempre apuntará al nodo que esté en la mitad. Una vez llena la lista, el primer valor que se insertará de nuevo en el árbol será el valor del medio. Luego se insertarán los demás
- Si la lista posee una longitud par el posicionado en la “mitad” estará en la posición $(\text{longitud} / 2) - 1$
- Si un subárbol que toca rebalancear tiene una altura menor o igual a 3 no se rebalancea
- Pasar los nodos del subárbol a rebalancear a una lista ordenada lo deben hacer de la manera más eficiente posible
- Cada nodo padre debe saber la cantidad de nodos de su subárbol
- Para facilitar la inserción de nodos en el árbol se utilizará lectura de archivos. El archivo, en cada línea, simplemente tendrá un número. Dicho número es el que se insertará en el árbol. Insertar todos los números en el árbol implica insertar todos los números del archivo
- Para ver el estado final del árbol deben tener un método `imprimir` que muestre todos los nodos del árbol junto con su profundidad y su cantidad de nodos

Ejemplo:

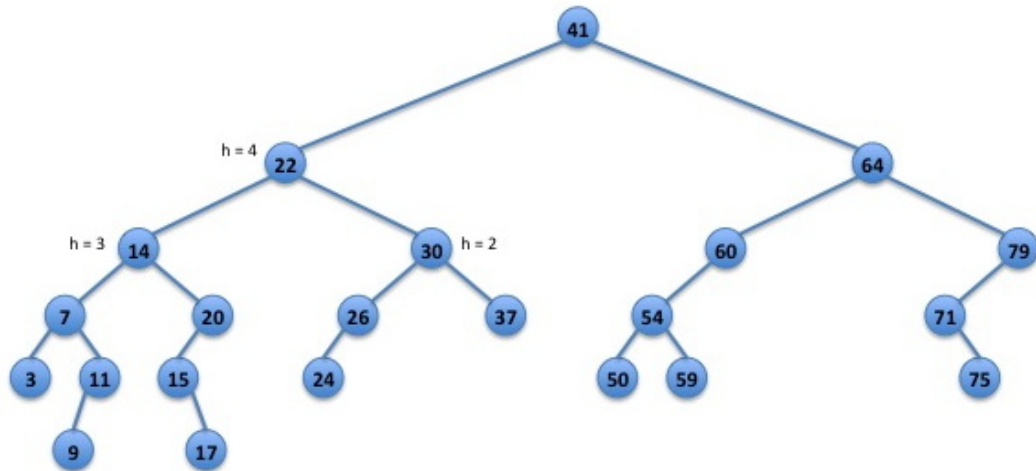


Ilustración 1 Este árbol presenta dos desbalances: Uno en el nodo 30 (El subárbol izquierdo tiene 2 nodos y el derecho tiene un nodo) y en el nodo 22 (el subárbol izquierdo tiene 8 nodos y el derecho tiene 4). El desbalance en el nodo 30 y en el nodo 14 se ignoran ya que ambos subárbol tienen alturas menores o iguales a 3. Por lo tanto, el desbalanceo se localiza en el nodo 22 con altura 4.

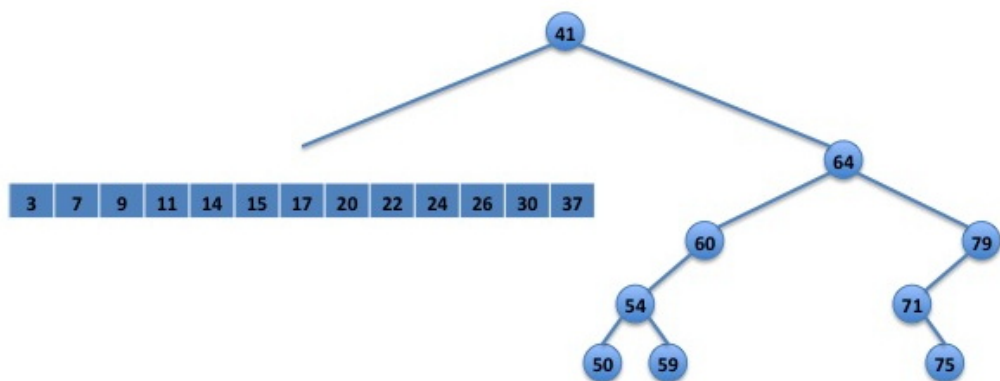


Ilustración 2 Todo el subárbol que tiene como padre el nodo 22 es convertido en un arreglo ordenado

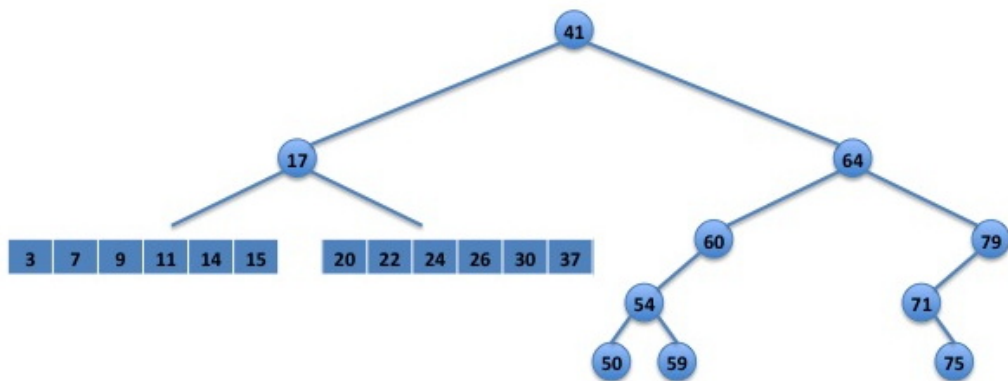


Ilustración 3 El primer nodo a insertar de nuevo es el nodo a la mitad del arreglo

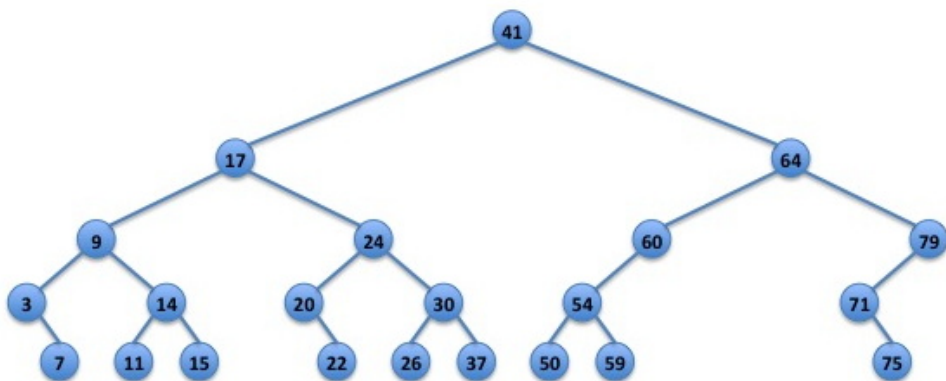


Ilustración 4 Por último, todos los nodos restantes son insertados de nuevo