

# Tarea Nueve

## Programación y Algoritmos I

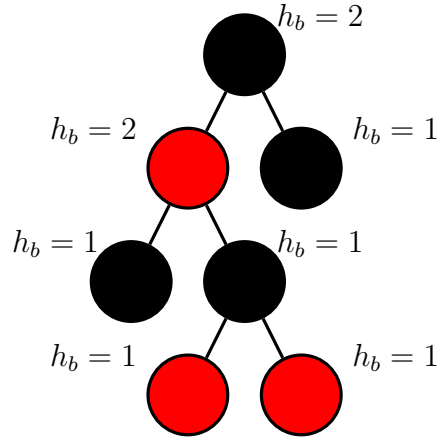
Andrea Quintanilla Carranza      Esteban Reyes Saldaña

18 de octubre de 2020

Un árbol **rojo-negro** es un ABB que tiene las siguientes restricciones

1. Cada nodo tiene un solo elemento (**label**) llamado color que es o rojo o negro.
2. La raíz del ABB es siempre negra.
3. Los hijos de un nodo rojo **tiene que ser ambos negros** ( o sea, no puede haber dos rojos consecutivos en un camino de la raíz a una hoja).
4. Las ligas vacías (apuntadores NULL en los nodos terminales) cuentan como negro.
5. Cualquier camino de un nodo  $v$  del árbol hacia un NULL tienen el mismo número de nodos negros (sin contar  $v$ ). Ese número se llama **altura negra** de  $v$  y lo notaremos  $h_b(v)$ .

En la figura siguiente, se ilustra un árbol rojo-negro con los valores  $h_b(v)$  correspondientes. No se ha representado los apuntadores NULL.



**Pregunta 1.** Demostrar que si  $r$  es la raíz de un árbol rojo-negro de altura  $h$ , tenemos

$$h_b(r) \geq \frac{h}{2}. \quad (1)$$

*Demostración.* Sea  $r$  la raíz de un árbol  $RB$  y  $h$  su altura. Sabemos que si existe un nodo rojo, por la propiedad 4, sus dos hijos son negros. Por lo que para cualquier camino, al menos la mitad de los nodos son negros. Luego para un camino de tamaño  $h$ , dado que  $h_b(r)$  está bien definida

$$h_b(r) \geq \frac{h}{2}.$$

□

**Pregunta 2.** Demostrar por inducción sobre la altura de los nodos que un subárbol enraizado en un nodo  $v$  tiene al menos  $2^{h_b(v)} - 1$  nodos internos.

*Demostración.*

- (i) Si  $h = 0$ , entonces el subárbol tiene un solo elemento  $v$  y corresponde a un nodo terminal, entonces  $h_b(v) = 1$ , ya que apunta a NULL. Así que

$$2^1 - 1 = 2^0 - 1 = 0.$$

De donde se cumple que  $v$  tiene al menos cero nodos internos.

- (ii) Tomemos ahora un árbol enraizado en un nodo  $v$  y altura  $h > 0$  y supongamos que  $v$  tiene dos hijos  $vc_1$  y  $vc_2$ . Notemos que la altura negra de sus hijos es  $h_b(v)$  o  $h_b(v) - 1$ , dependiendo del color que tengan dichos hijos. Sin pérdida de generalidad, supongamos que la altura negra de los hijos es  $h_b(v) - 1$ .

Supongamos entonces que la propiedad se cumple para todo subárbol de altura menor que  $k$ . Particularmente, sabemos que la altura de los subárboles enraizados en  $vc_1$  y  $vc_2$  es menor a la altura del subárbol enraizado en  $v$ , entonces

- el subárbol enraizado en  $vc_1$  tiene al menos  $2^{h_b(v)-1} - 1$  nodos y
- el subárbol enraizado en  $vc_2$  tiene al menos  $2^{h_b(v)-1} - 1$  nodos.

así que el subárbol enraizado en el padre  $v$  tendrá al menos

$$2^{h_b(v)-1} - 1 + 2^{h_b(v)-1} - 1 + 1 = 2(2^{h_b(v)-1}) - 1 = 2^{h_b(v)} - 1$$

nodos, así que la propiedad se cumple también para todo subárbol de tamaño  $k$ , por el principio de inducción matemática, tenemos que la propiedad se cumple para toda  $h \in \mathbb{N}$ .

□

**Pregunta 3.** Deducir de lo anterior que, si  $n$  es el número total de nodos, la altura  $h$  del árbol satisface:

$$h \leq 2 \log_2(n + 1)$$

**Solución.** Sea  $h$  la altura de un árbol RB y  $r$  su raíz. Por el ejercicio anterior sabemos que  $r$  tiene al menos  $2^{h_b(r)} - 1$  nodos. Es decir, si  $n$  es el total de nodos,

$$n \geq 2^{h_b(r)} - 1.$$

Por (1) tenemos que

$$\begin{aligned} h_b(r) &\geq h/2 \\ 2^{h_b(r)} &\geq 2^{h/2} \\ n &\geq 2^{h/2} - 1 \\ n + 1 &\geq 2^{h/2} \\ \log_2(n + 1) &\geq h/2 \\ 2 \log_2(n + 1) &\geq h. \end{aligned}$$

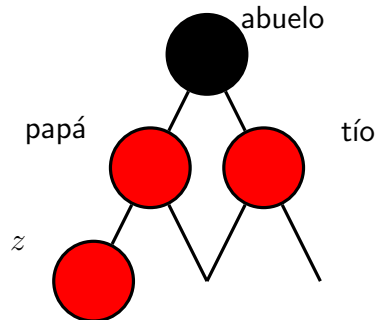
**Pregunta 4.** Definimos la inserción de un nuevo dato en un árbol rojo-negro como sigue: Insertamos el nuevo nodo  $w$  como en un ABB normal (bajando hacia su lugar por búsqueda) y lo coloreamos como **rojo**. Si ese nodo es la raíz ( $w$  fue el primer nodo), lo coloreamos como negro. Mostrar que el único caso en que se puede generar una violación de las reglas de árbol rojo-negro es cuando el padre de  $w$  es rojo. **Solución.** Sea  $z$  un nuevo nodo insertado a un árbol RB. Enlistando las posibles fallas de las propiedades de un árbol rojo-negro tenemos

1. Siempre se cumple.
2. Si después de hacer inserción  $z$  es la raíz, esta es roja, por lo que se viola esta propiedad. Así que se recolorea a negro. Las propiedades 1,2,3,4 se siguen cumpliendo por defecto y la propiedad 5 también, dado que  $z$  no cuenta para calcular la altura negra.
3. Esta propiedad falla si el padre de  $z$  es rojo.
4. Se sigue cumpliendo.
5. Se cumple, dado que antes de insertar  $z$  el futuro padre de  $z$  apunta a NULL y al insertar  $z$  se le resta 1 a la altura negra de los árboles que estén en ese camino. Pero ahora  $z$  es rojo y apunta a NULL, así que se suma 1 a la altura negra de los nodos en esa rama.

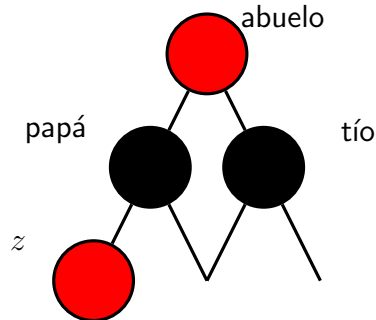
Concluimos que la única propiedad que se puede violar al insertar un nodo es cuando el padre de  $z$  es rojo.

**Pregunta 5.** Mostrar que, en el caso anterior de violación, si el nodo tío de  $w$  (es decir, el otro hijo de su abuelo) es también rojo, hay una corrección muy simple que se puede hacer al *cambiar de colores el abuelo, el papá y el tío*. ¿Cómo cambia la altura negra de los nodos del árbol con esta corrección? Mostrar que la corrección puede provocar una violación al nivel el abuelo. ¿Cuál es la complejidad de esta corrección?

**Solución.** Supongamos que se inserta un nodo  $z$  con color rojo y que su padre también es rojo. Supongamos además que el tío de  $z$  también es rojo. Notemos que, dado que el árbol es RB, el abuelo de  $z$  no puede ser rojo (porque entonces sus hijos serían rojos y eso viola la propiedad 4). Así que el abuelo de  $z$  es negro.



Si cambiamos los colores del abuelo, el papá y el tío ahora tendríamos la configuración



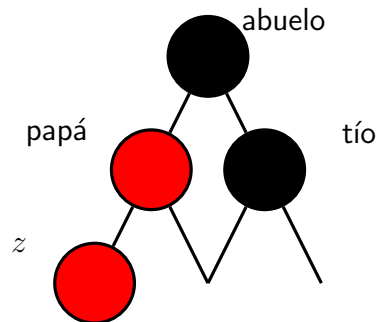
Notemos que si la nueva configuración es un árbol rojo-negro, la altura negra de los nodos asociado a esa rama aumentará en 1.

Si el padre del abuelo es negro, entonces el árbol sigue siendo rojo-negro. Si el padre del abuelo es rojo, se hace el mismo cambio a nivel del abuelo tantas veces como violaciones haya en esa rama.

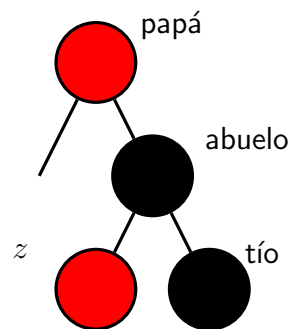
En el mejor caso, solo se hace una corrección y esta se hace en  $O(1)$ . En el peor caso, esta corrección se hace hasta llegar a la raíz, en este caso, la complejidad será proporcional a la altura del árbol entonces tenemos  $O(\log(n))$ .

**Pregunta 6.** Mostrar que en el otro caso (si el tío es negro), se puede usar las mismas **rotaciones** que vimos en el caso de árboles AVL para corregir el árbol rojo-negro. ¿Cómo cambia la altura negra de los nodos del árbol con esta corrección? ¿Cuál es la complejidad de esta corrección?

**Solución.** Sin pérdida de generalidad supongamos que dicho árbol tiene estructura



al aplicar la rotación obtendremos



Así este subárbol es rojo-negro. Si el papá del abuelo es negro, terminamos con la corrección. Si el papá del abuelo es rojo,