

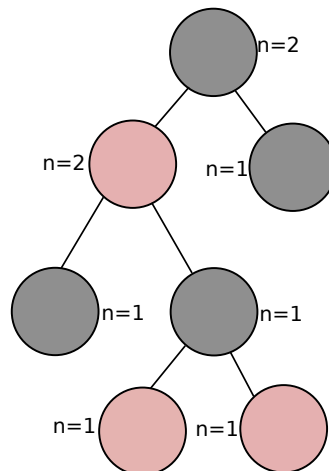
# Programación y Algoritmos I

## Tarea 7: Árboles rojos-negros

En la clase vimos que una clase de árboles binarios de búsqueda (ABBs) balanceados son los **árboles AVL**. En esta tarea, estudiaremos otra clase de ABBs llamados **árboles rojos-negros**. Un árbol rojo-negro es un ABB que tiene las siguientes restricciones:

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

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



### Pregunta 1 [ 0.5 puntos ]

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

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

### Pregunta 2 [ 1 puntos ]

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.

**Pregunta 3** [ 0.5 puntos ]

Deducir de lo anterior que la altura  $h$  del arbol satisface:

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

**Pregunta 4** [ 0.5 puntos ]

Vamos a definir la inserción de un nuevo dato en un arbol rojo-negro de la manera siguiente: Insertamos el nuevo nodo  $w$  como en un BST normal (bajando hacia su lugar por búsqueda) y lo coloreamos como **rojo**. Si ese nodo es la raiz ( $w$  fue el primer nodo), lo coloreamos como negro. Mostrar que el único caso en que se puede generar una violación de las reglas de arbol rojo-negro es cuando el padre de  $w$  (si hay) es **rojo**.

**Pregunta 5** [ 1 puntos ]

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 papa y el tío**. Cómo cambia la altura negra de los nodos del árbol con esta corrección?Cuál es la complejidad de esta corrección?

**Pregunta 6** [ 1 puntos ]

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 arbol rojo-negro. Cómo cambia la altura negra de los nodos del árbol con esta corrección?Cuál es la complejidad de esta corrección?

**Pregunta 7** [ 5.5 puntos ]

Implementar la estructura de árbol rojo-negro, y sus funciones de búsqueda de una llave y de inserción de una nueva llave.