ÁRBOLES RN

Árboles Rojo-Negro

Jose Antonio Castro
Anita Marti

Menti

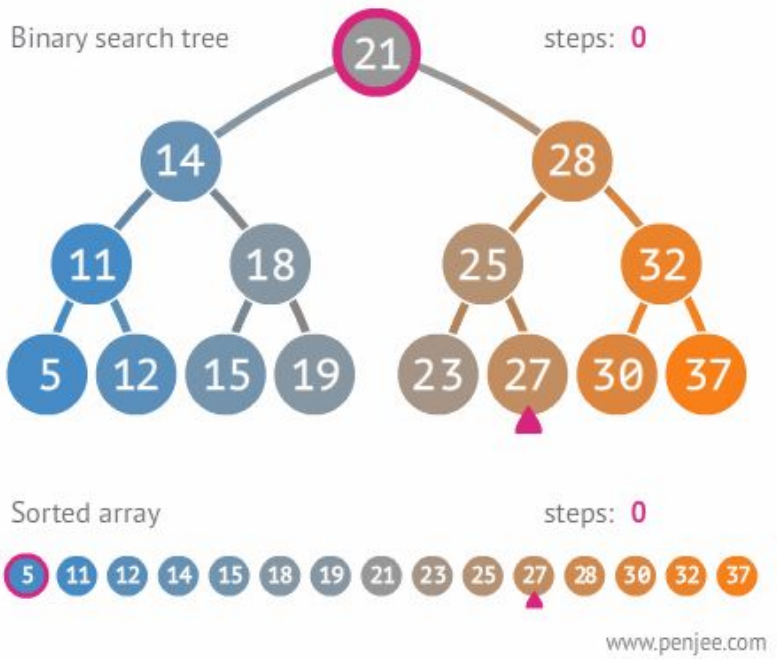
2291 6260

Pregunta

- ¿Cual es la principal ventaja de *cualquier* árbol de búsqueda?

Pregunta

- ¿Cual es la principal ventaja de *cualquier* árbol de busqueda?



Pregunta

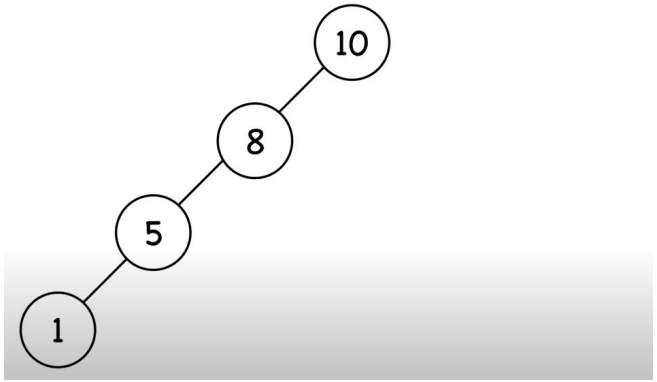
- ¿Cual es la principal ventaja de *cualquier* árbol de busqueda?

- ¿Es siempre un árbol de búsqueda una buena opción?

Pregunta

- ¿Cual es la principal ventaja de *cualquier* árbol de busqueda?

- ¿Es siempre un árbol de búsqueda una buena opción?



Pregunta

- ¿Cual es la principal ventaja de *cualquier* árbol de busqueda?
- ¿Es siempre un árbol de búsqueda una buena opción?
- Que ventaja nos otorga usar un arbol balanceado sobre un BST

Contenidos a revisar

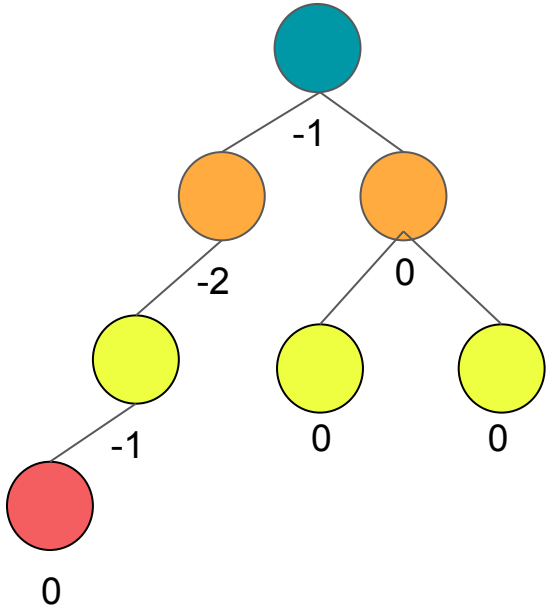
Rojo-Negro

Repaso AVL

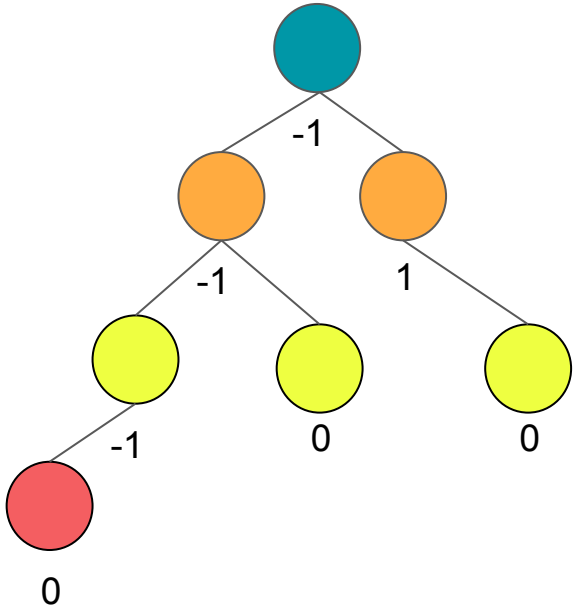
Propiedades:

- Es un tipo de binary search tree (BST)
- Está balanceado
- Las alturas de los hijos de la raíz difieren a lo más de 1 entre ellos
- Cada hijo es AVL
- Altura es log(n)

No Balanceado



Balanceado



Árbol Rojo-Negro

Es un tipo de AVL que cumple las siguientes propiedades:

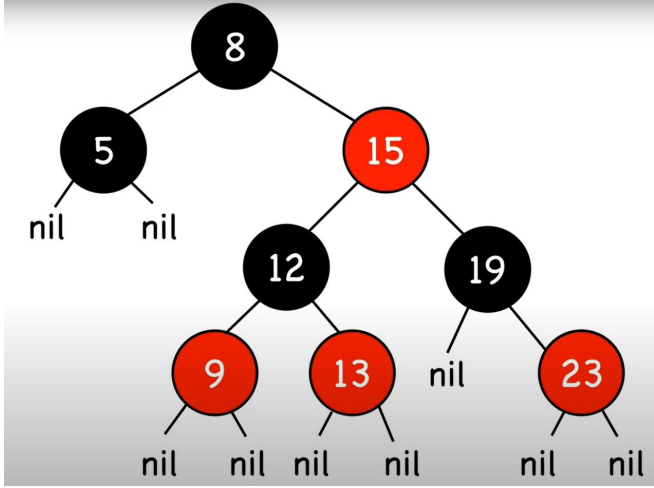
1. Cada nodo es **rojo** o **negro**

2. La raíz del árbol es negra

3. Si un nodo es **rojo**, entonces sus hijos deben ser **negros**

4. La cantidad de nodos **negros** en el camino a cada hoja debe ser la misma

Árbol Rojo-Negro



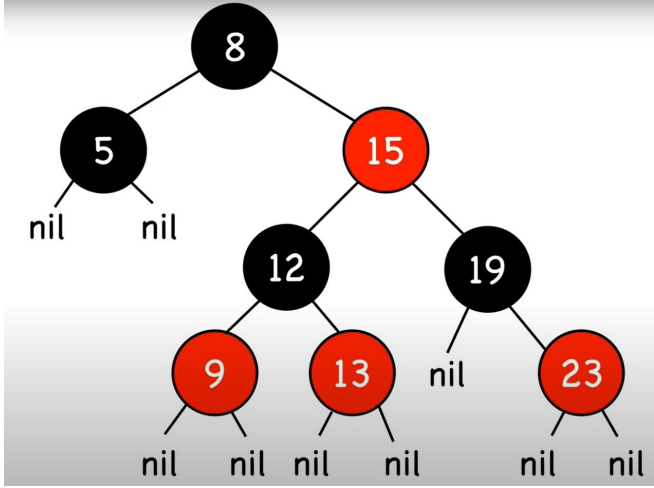
Para la inserción:

- Los nodos siempre se insertan como **rojos**
- Si su padre es **rojo** ocurre una de dos cosas:

1. Si su tío es **negro**, se tiene un aumento de grado en el nodo 2-4. Esto se soluciona con rotaciones y cambios de color. (Los nodos nulos se consideran negros)

2. Si su tío es **rojo**, se da el caso en que 2-4 rebalsa. Esto se soluciona cambiando la coloración, lo que puede generar el mismo caso hacia arriba

Árbol Rojo-Negro



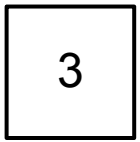
Para la inserción:

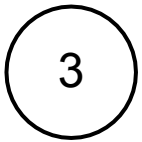
- Los nodos siempre se insertan como **rojos**
- Si su padre es **rojo** ocurre una de dos cosas:

1. Si su tío es **negro**, se tiene un aumento de grado en el nodo 2-4. Esto se soluciona con rotaciones y cambios de color

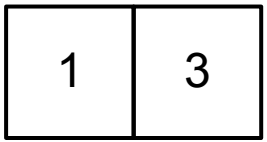
2. Si su tío es **rojo**, se da el caso en que 2-4 rebalsa. Esto se soluciona cambiando la coloración, lo que puede generar el mismo caso hacia arriba

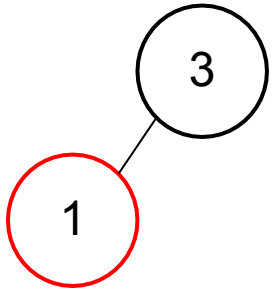
Rojo negro: Inserte los siguientes datos en el árbol: 3, 1, 7, 8, 9, 2, 10, 0, 4, 5, 6, -1



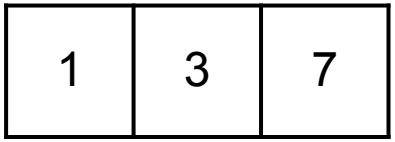


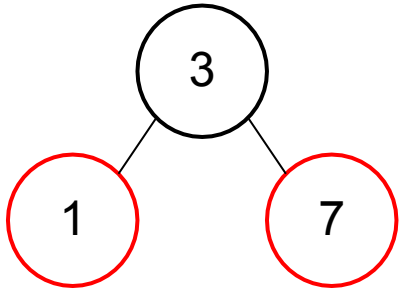
Rojo negro: Inserte los siguientes datos en el árbol: 1, 7, 8, 9, 2, 10, 0, 4, 5, 6, -1



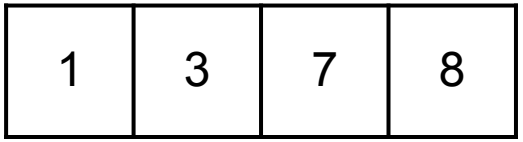


Rojo negro: Inserte los siguientes datos en el árbol: 7, 8, 9, 2, 10, 0, 4, 5, 6, -1

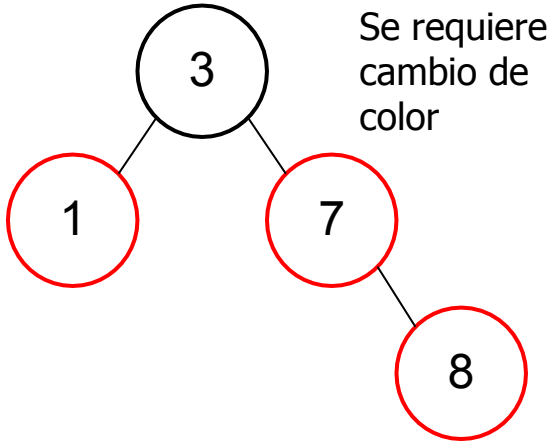




Rojo negro: Inserte los siguientes datos en el árbol: 8, 9, 2, 10, 0, 4, 5, 6, -1

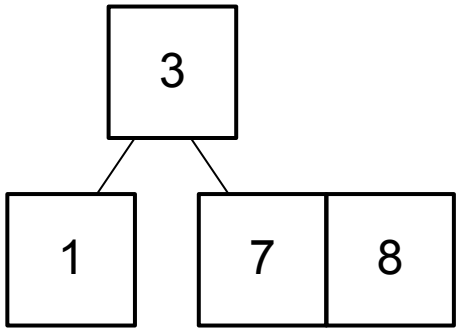


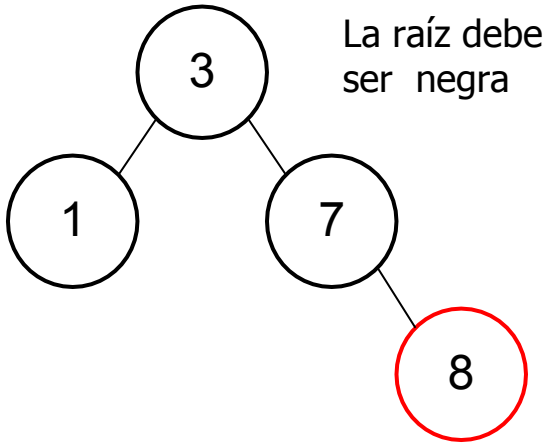
Se debe equilibrar el nodo



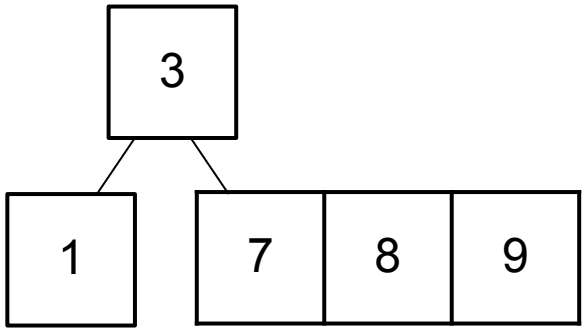
Se requiere cambio de color

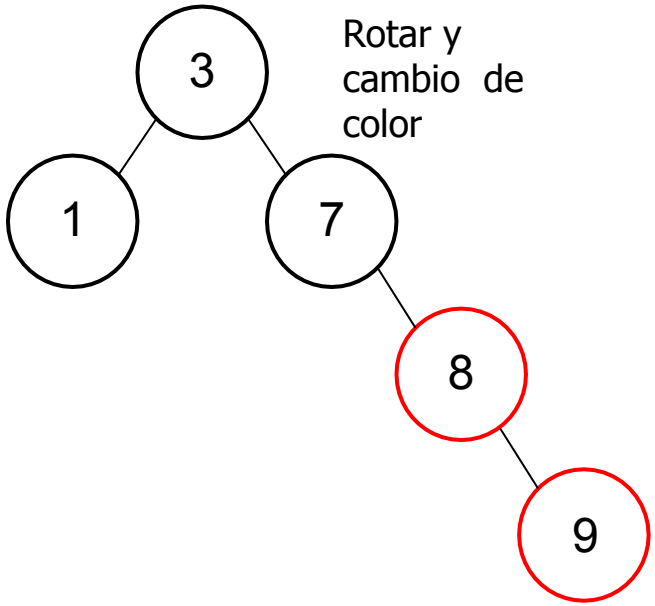
Rojo negro: Inserte los siguientes datos en el árbol: 8, 9, 2, 10, 0, 4, 5, 6, -1



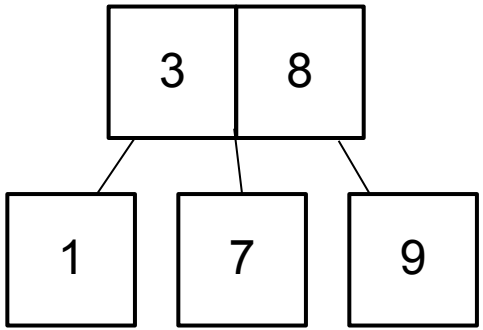


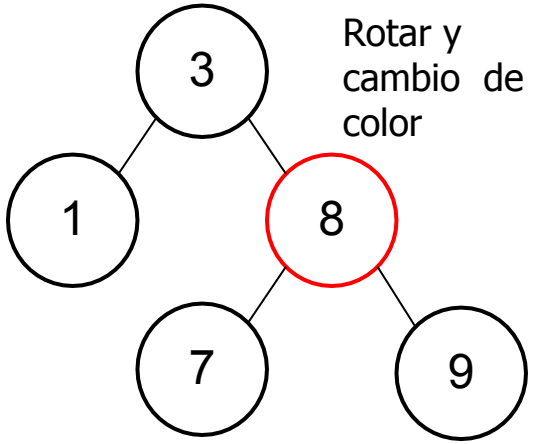
Rojo negro: Inserte los siguientes datos en el árbol: 9, 2, 10, 0, 4, 5, 6, -1



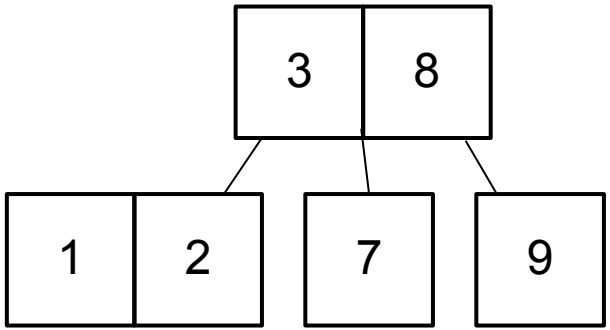


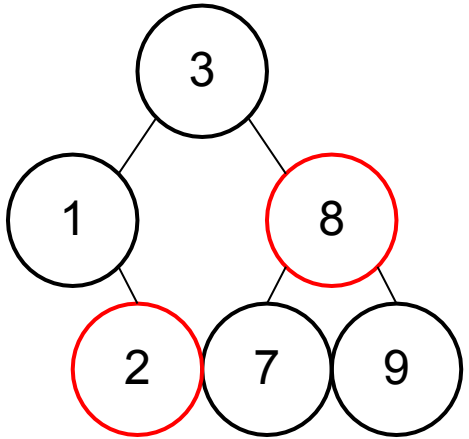
Rojo negro: Inserte los siguientes datos en el árbol: 9, 2, 10, 0, 4, 5, 6, -1



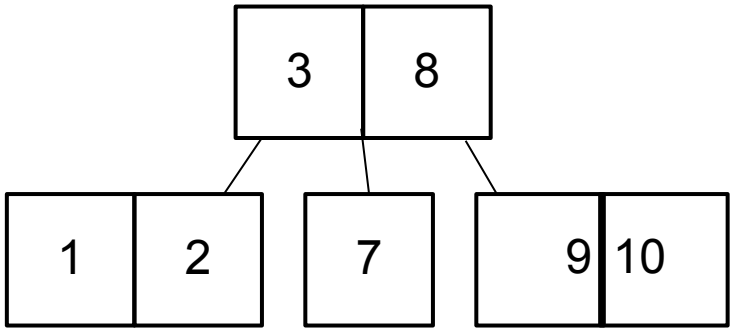


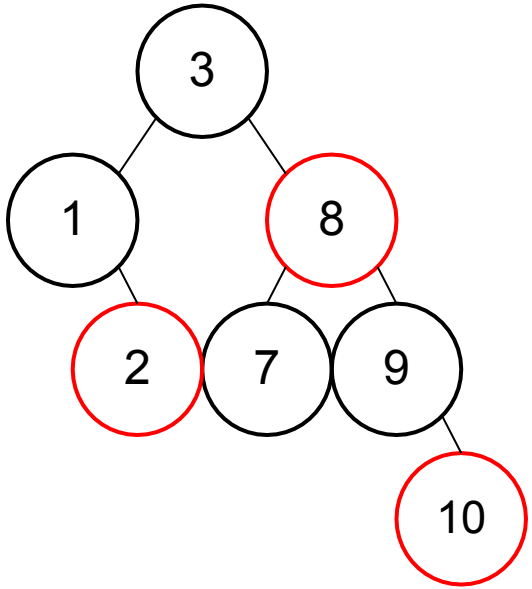
Rojo negro: Inserte los siguientes datos en el árbol: 2, 10, 0, 4, 5, 6, -1



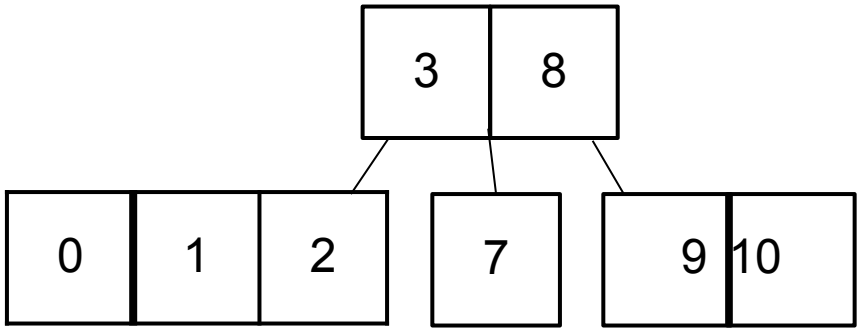


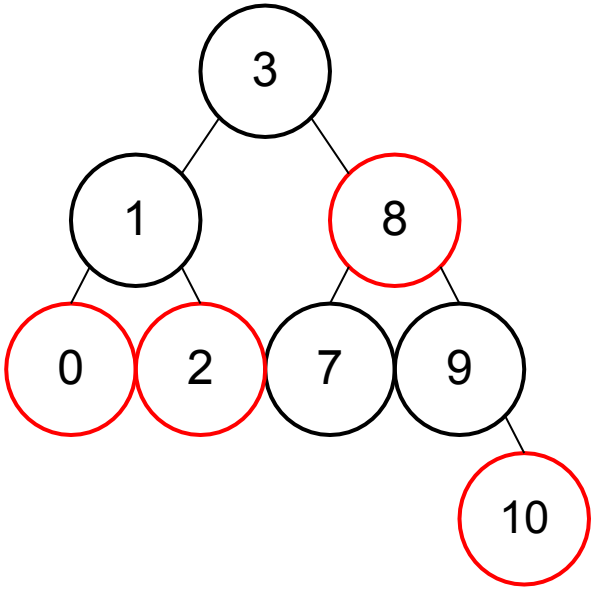
Rojo negro: Inserte los siguientes datos en el árbol: 10, 0, 4, 5, 6, -1



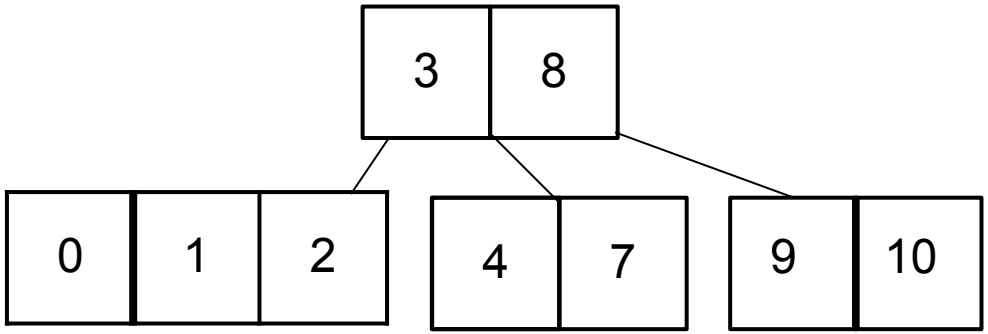


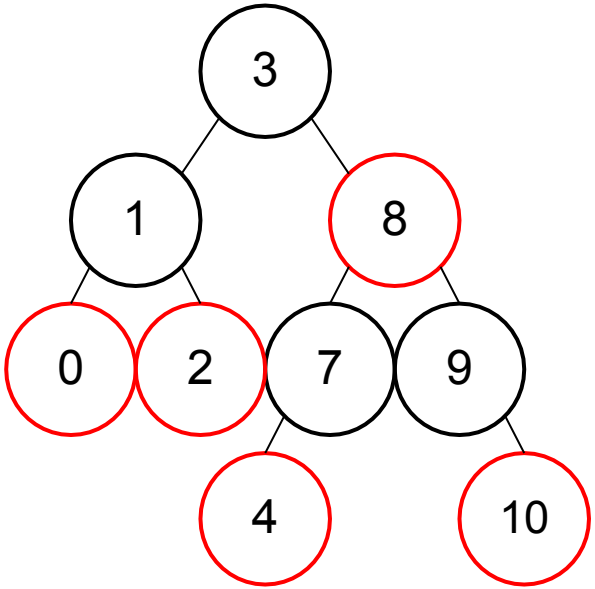
Rojo negro: Inserte los siguientes datos en el árbol: 0, 4, 5, 6, -1



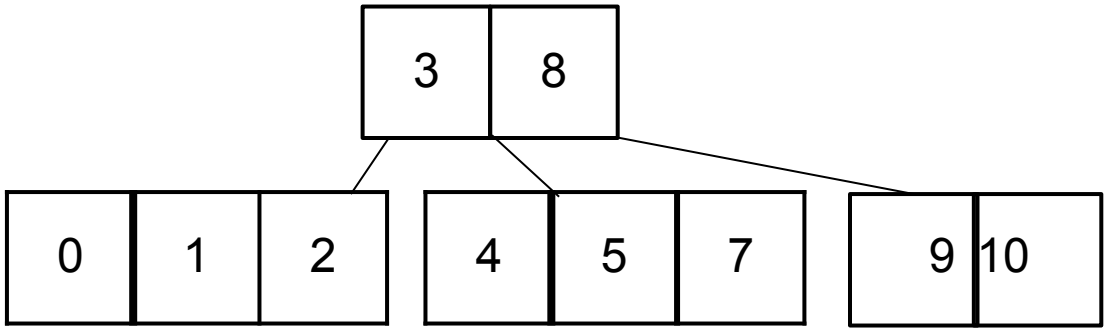


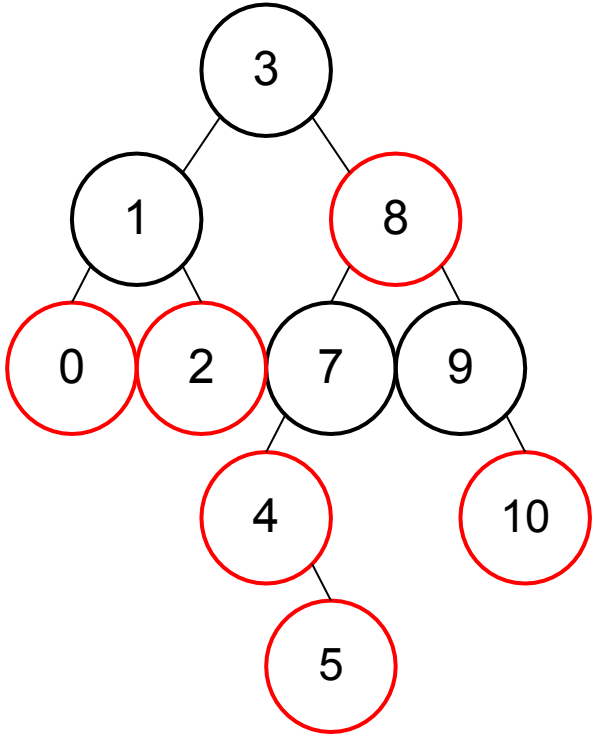
Rojo negro: Inserte los siguientes datos en el árbol: 4, 5, 6, -1



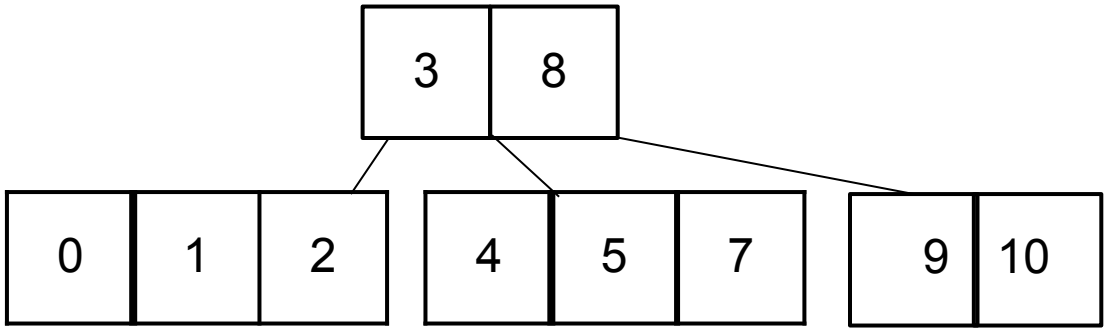


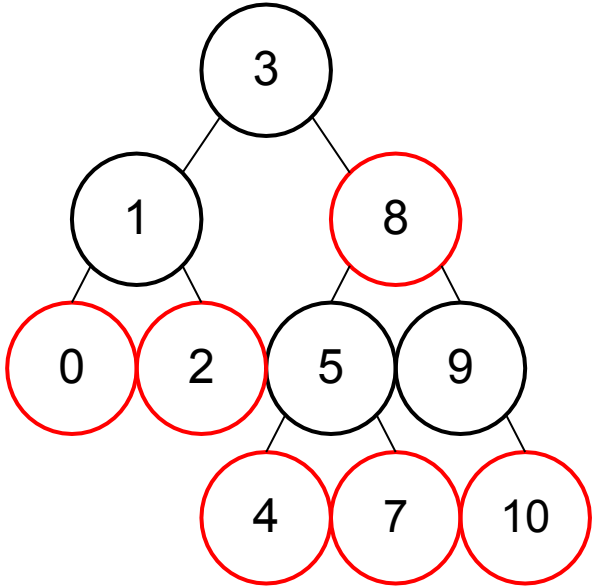
Rojo negro: Inserte los siguientes datos en el árbol: 5, 6, -1



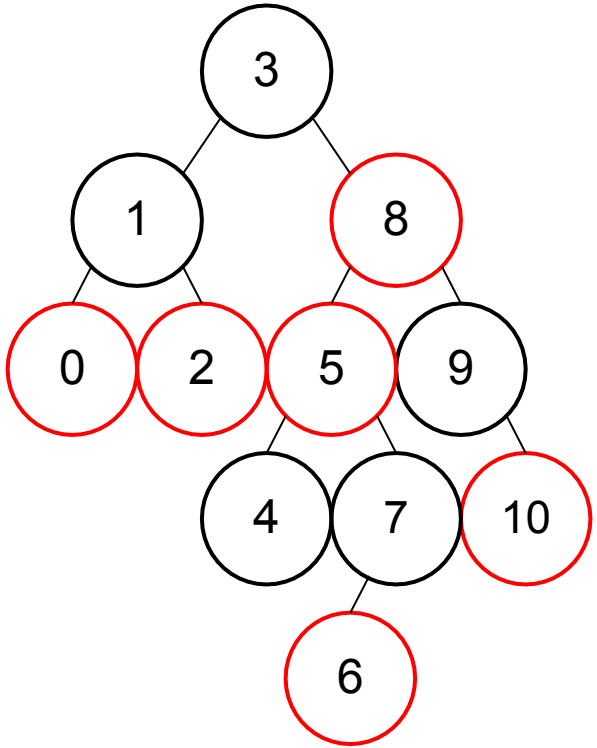


Rojo negro: Inserte los siguientes datos en el árbol: 5, 6, -1

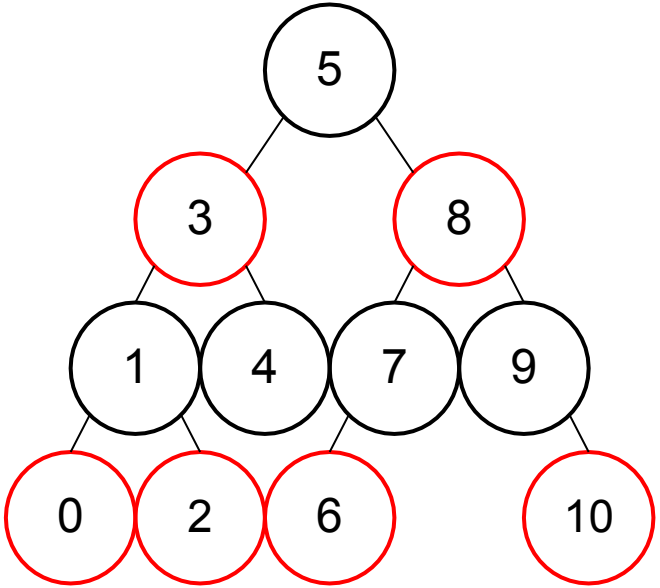




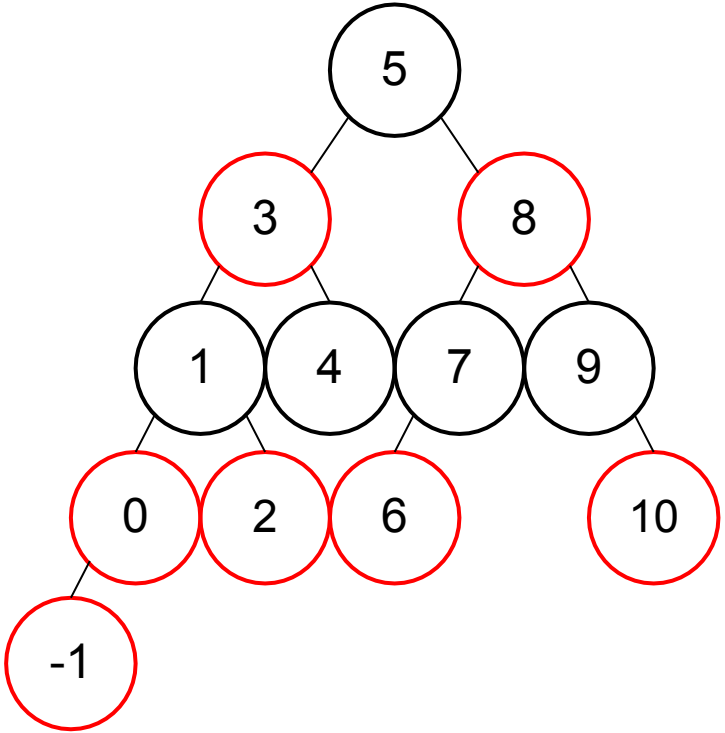
Rojo negro: Inserte los siguientes datos en el árbol: 6, -1



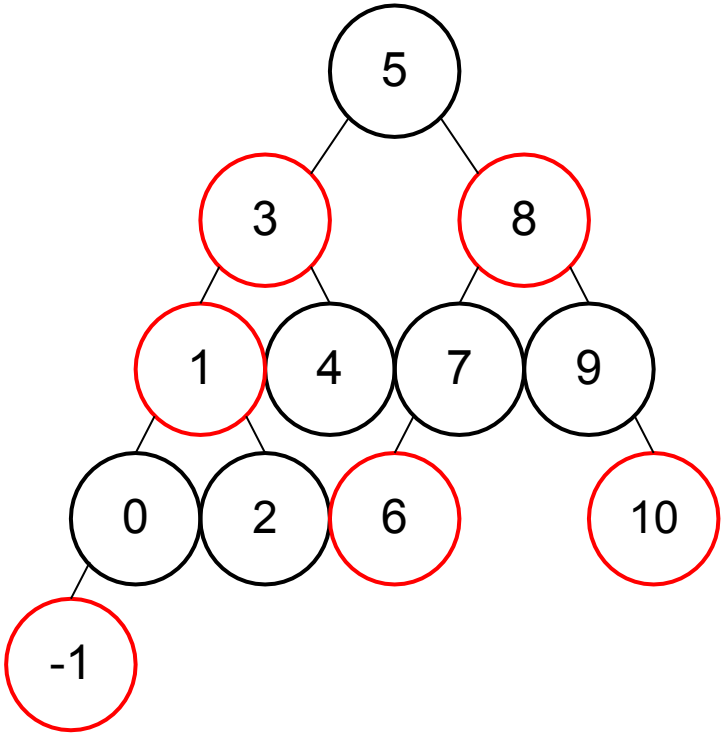
Rojo negro: Inserte los siguientes datos en el árbol: 6, -1



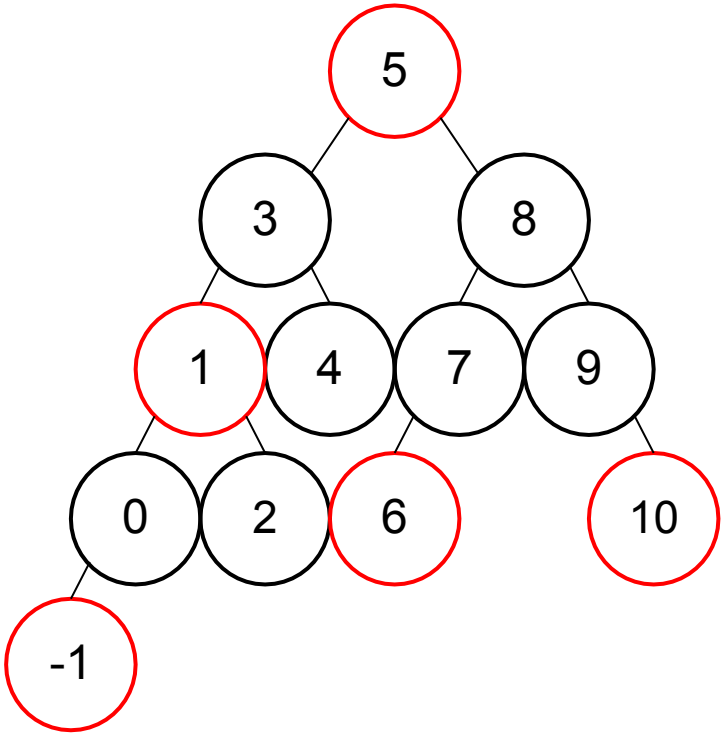
Rojo negro: Inserte los siguientes datos en el árbol: -1



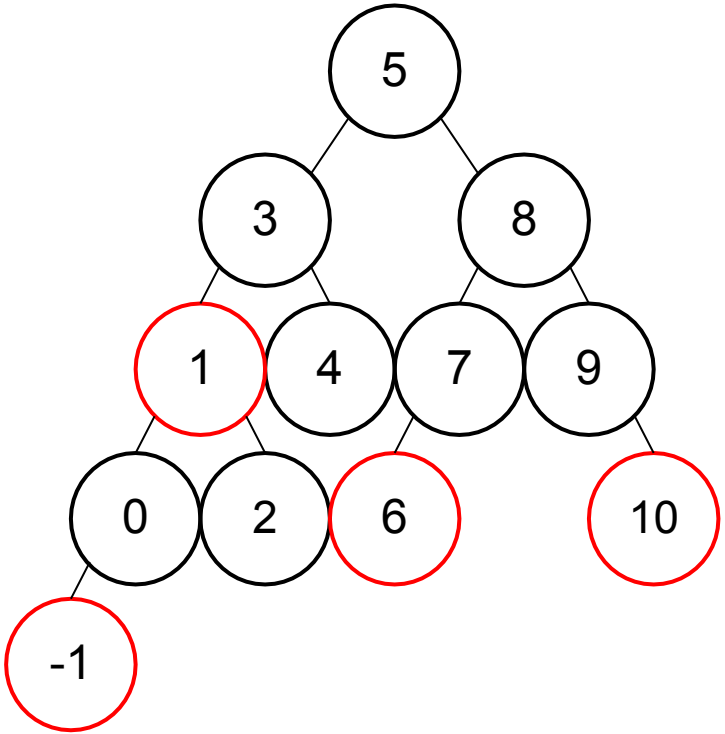
Rojo negro: Inserte los siguientes datos en el árbol: -1



Rojo negro: Inserte los siguientes datos en el árbol: -1



Rojo negro: Inserte los siguientes datos en el árbol: -1



Árboles Rojo-Negro:

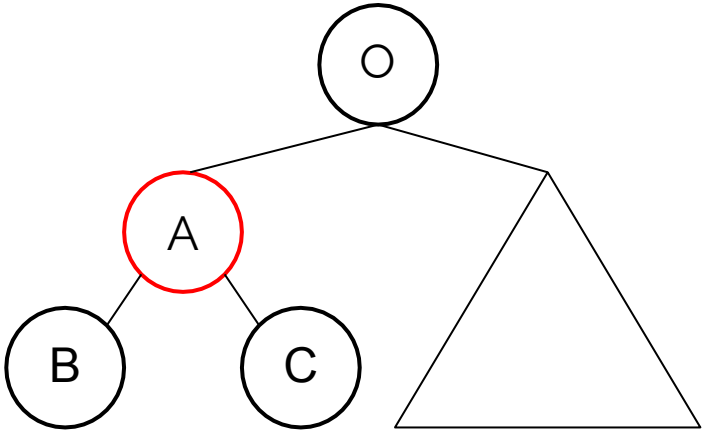
Considera un árbol rojo-negro en que el número de nodos negros en cada ruta de la raíz a una hoja es k

a. ¿Cuál es la altura máxima posible del árbol? ¿Y la mínima?

b. ¿Cuál es el máximo número de nodos que puede tener el árbol? ¿Y el mínimo?

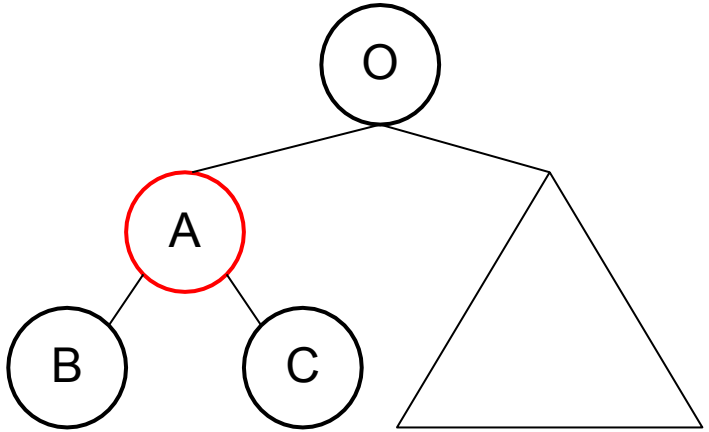
¿Cuál es la altura máxima posible del árbol? ¿Y la mínima?

Recordamos la propiedad del arbol Rojo-Negro, donde nescesariamente los hijos de un nodo rojo han de ser negros



¿Cuál es la altura máxima posible del árbol? ¿Y la mínima?

Recordamos la propiedad del arbol Rojo-Negro, donde nescesariamente los hijos de un nodo rojo han de ser negros

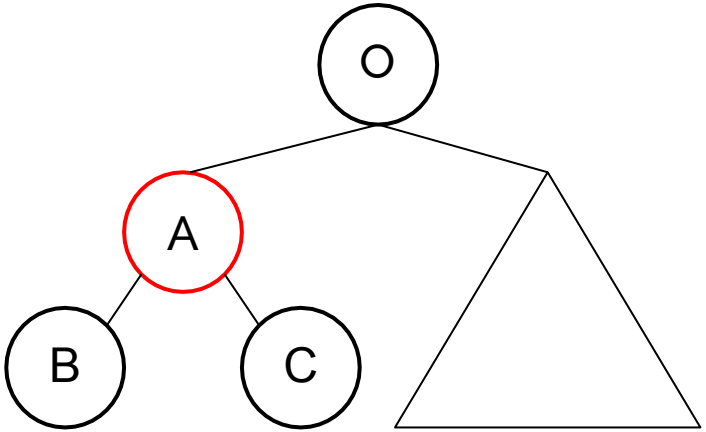


Por lo tanto, a lo más habrán *k* nodos rojos de profundidad, dado que para cada nodo rojo de profundidad, hay *k* nodos negros que pueden ser su padre

Luego, La altura máxima del árbol es de *2k*

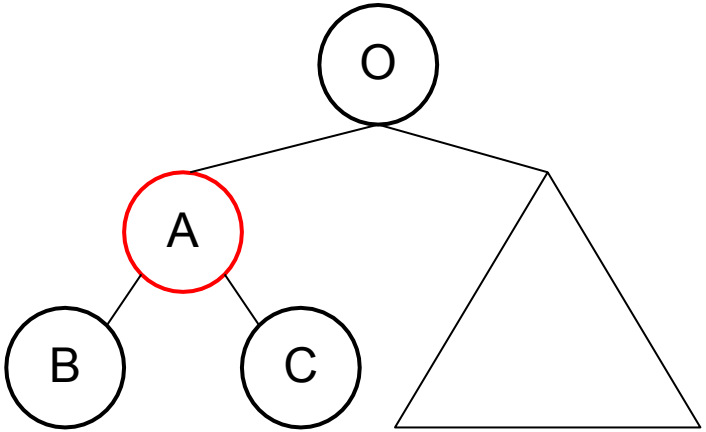
¿Cuál es la altura máxima posible del árbol? ¿Y la mínima?

Recordamos la propiedad del arbol Rojo-Negro, donde nescesariamente los hijos de un nodo rojo han de ser negros



¿Cuál es la altura máxima posible del árbol? ¿Y la mínima?

Recordamos la propiedad del arbol Rojo-Negro, donde nescesariamente los hijos de un nodo rojo han de ser negros



A su vez, la altura minima la alcanzariamos en un arbol binario compuesto de solo nodos negros, y por lo tanto, la altura seria de *k*

¿Cuál es el máximo número de nodos que puede tener el árbol?

¿Y el mínimo?

¿Cuál es el máximo número de nodos que puede tener el árbol?

¿Y el mínimo?

Recordando la parte anterior, definimos que la altura mínima de nuestro árbol es de *k*, y por lo tanto al saber que nuestro árbol es AVL, concluimos que el número máximo de nodos es de $2^k - 1$

$$2^h - 1 = 2^{2k} - 1$$

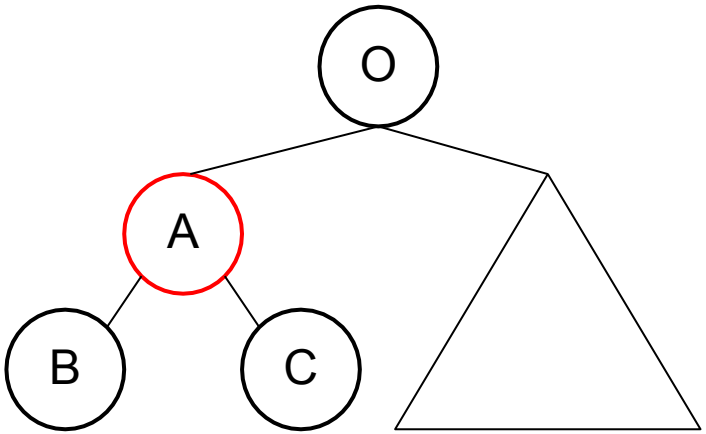
Ahora, en el contexto de la inserción de un nuevo nodo

(a) ¿Cuál es la cantidad máxima de cambios de color que pueden ocurrir?

(b) ¿Cuál es la cantidad máxima de rotaciones (simples o dobles) que pueden ocurrir?

(a) ¿Cuál es la cantidad máxima de cambios de color que pueden ocurrir?

Recordamos la propiedad del arbol Rojo-Negro, donde nescesariamente los hijos de un nodo rojo han de ser negros

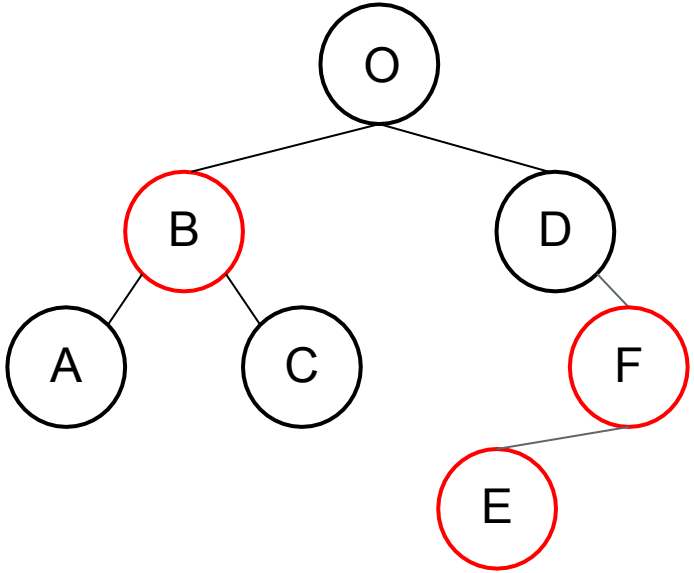


Si es que quisiéramos insertar un nodo rojo al árbol de altura *2k*. Hemos de realizar un cambio de color por cada nivel en la ruta de insercion

Por lo que es del orden de O(k) cambios de color

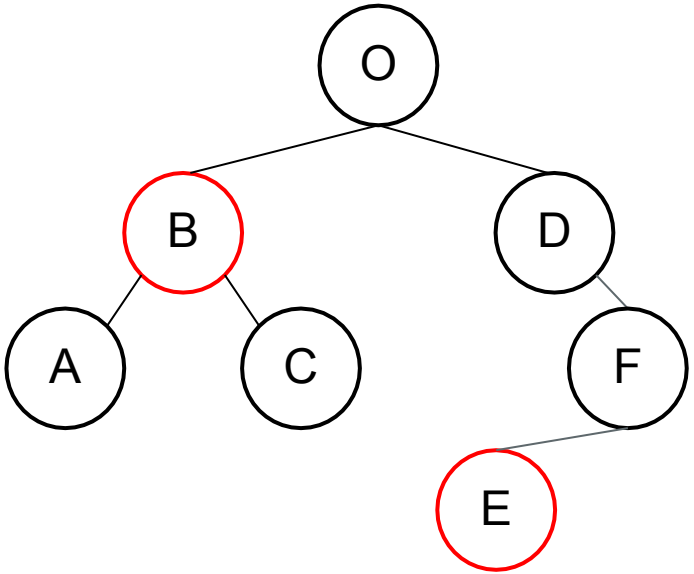
(b) ¿Cuál es la cantidad máxima de rotaciones (simples o dobles) que pueden ocurrir?

Una rotación corresponde reordenar los nodos, dado que todos los datos nuevos inicialmente tratamos de insertarlos como hojas, por lo que la rotación ocurre solo en la hoja donde se insertó dicho nodo y por lo tanto solo puede ocurrir una rotación por inserción



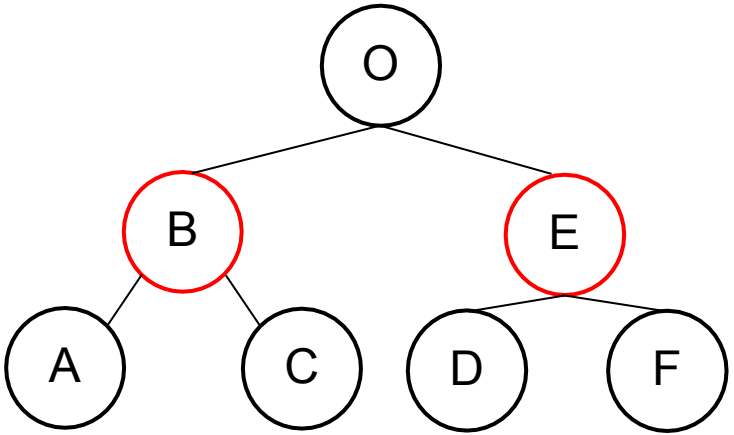
Un nodo rojo no puede tener un hijo rojo, se cambia color

(b) ¿Cuál es la cantidad máxima de rotaciones (simples o dobles) que pueden ocurrir?



La Cantidad de nodos negros a una hoja es diferente, por lo que se rota

(b) ¿Cuál es la cantidad máxima de rotaciones (simples o dobles) que pueden ocurrir?



Pregunta Final (Con Premio)

¿Cual es la ventaja principal de utilizar un Arbol Rojo Negro (RBT) por sobre un AVL?

(Recordar que la complejidad para ambos árboles es log(n) para insertar, eliminar y buscar)

Hint: Pensar en términos y de que operaciones se dejan de hacer

Pregunta Final (Con Premio)

¿Cual es la ventaja principal de utilizar un Arbol Rojo Negro (RBT) por sobre un AVL?

R. Las eliminaciones son más caras en un AVL. Ya que en caso de eliminar algo (log(n)) luego se deben propagar hasta el root en C rotaciones, por lo que en la realidad serían CLog(N) operaciones. Sin embargo RBT asegura que se ejecutarán solo log(n) pasos

(Lo mismo para la inserción)

En resumen, un AVL es mucho mejor para buscar ya que es *estricto* con su balance. Pero el RBT es flexible por lo que insert-delete es más rápido (no hay factor C)

STUART LITTLE IS A RAT=

CANCEL STUART LITTLE

I fucking hate Stuart Little. I know what you're thinking, this is some kind of funny joke, but no. Stuart Little is a piece of shit. A damn rat got picked over actual children at an orphanage and he's supposed to be a hero? And I can't even tell you how many damn times I've seen a great parking space only to turn the corner and realise Stuart Little is already parked there in his stupid little fucking convertible. He took my wife and the kids and my house and my job. I swear to fucking god, I'm going to kill myself and take that goddamn rodent to hell with me. Stuart Little has ruined my family. Last summer, I approached the miserable mouse in the street, and asked him for his autograph, because my son is a huge fan. The fucking rat gave me the autograph and told me to burn in hell. Later, when I gave my son the autograph he started crying and said he hated me. Turns out the mousefucker didnt write his autograph, no, he wrote "you're a piece of shit, and i fucked your mom". I'm now divorced, and planning a huge class-action lawsuit against the white devil that ruined my life. Your time is almost over, Stuart. All the people you've wronged will rise against you.