

www.koalastothemax.com

ÁRBOLES

Árboles y Árboles Binarios

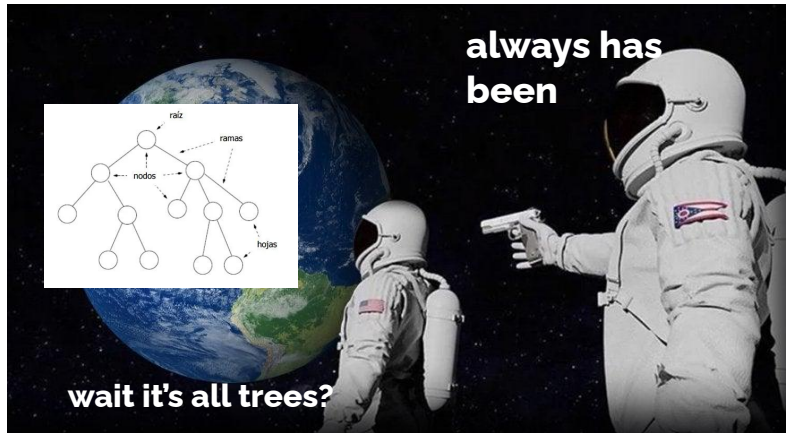
José Antonio Castro
Pablo Soto

Contenidos a revisar

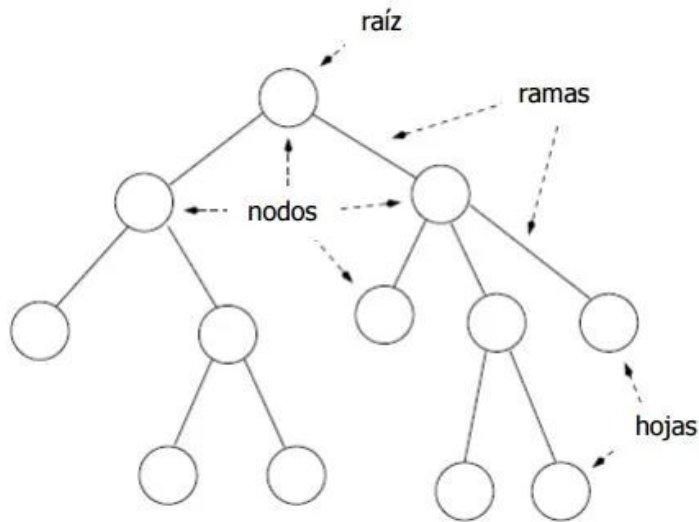
BST

-

AVL

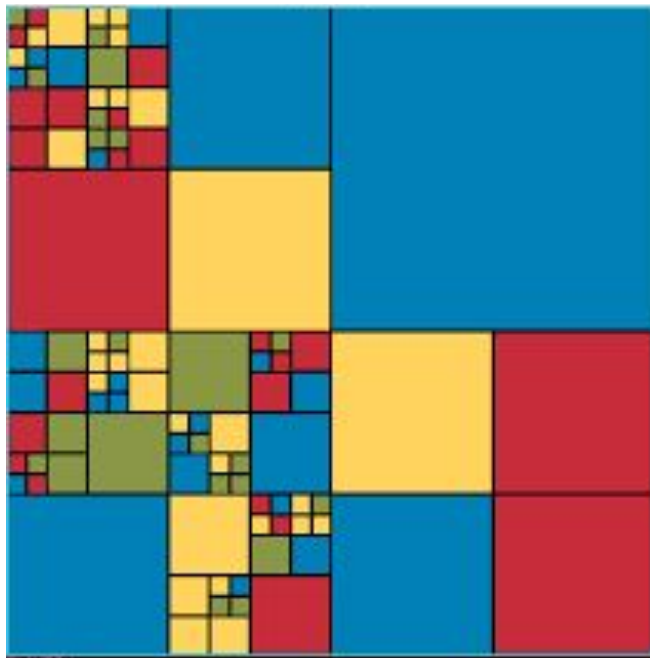


Que es un árbol



- Estructura de datos con nodos
- Tiene una raíz
- Cada nodo puede tener Cero o más hijos
- Cada nodo tiene un padre (menos el nodo raíz)
- No hay ciclos
- Se arman con normas pre-establecidas

Un ejemplo Gráfico, Quad-Tree



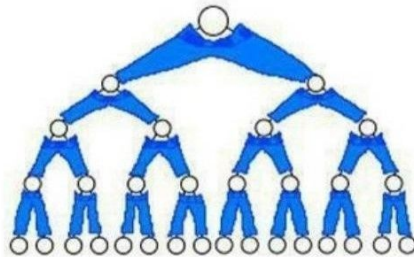
Binary Search Tree (BST)

Propiedades:

- Cada nodo tiene a lo más 2 hijos.
- Altura mínima de la rama más larga es $O(\log(n))$
- Altura máxima de $O(n)$, puede no estar balanceado
- Los hijos a la derecha de un nodo son todos mayores a dicho nodo mientras que los de la izquierda son todos menores

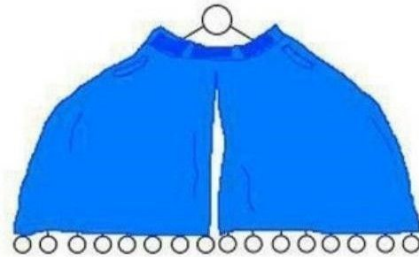
If a binary tree wore pants would he wear them

like this



or

like this?

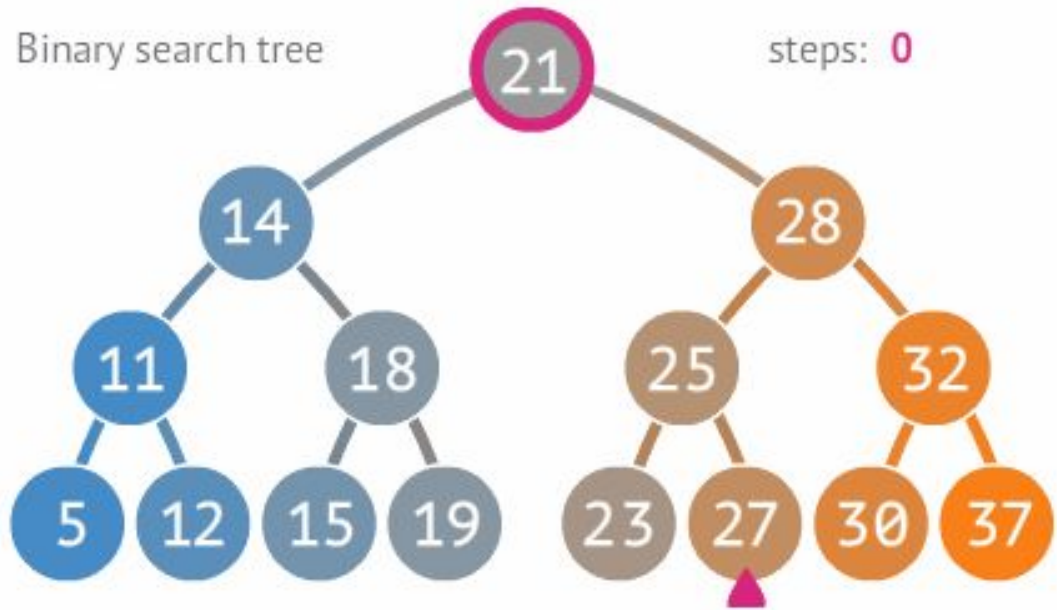


Árbol AVL

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 $\log(n)$

Árbol AVL



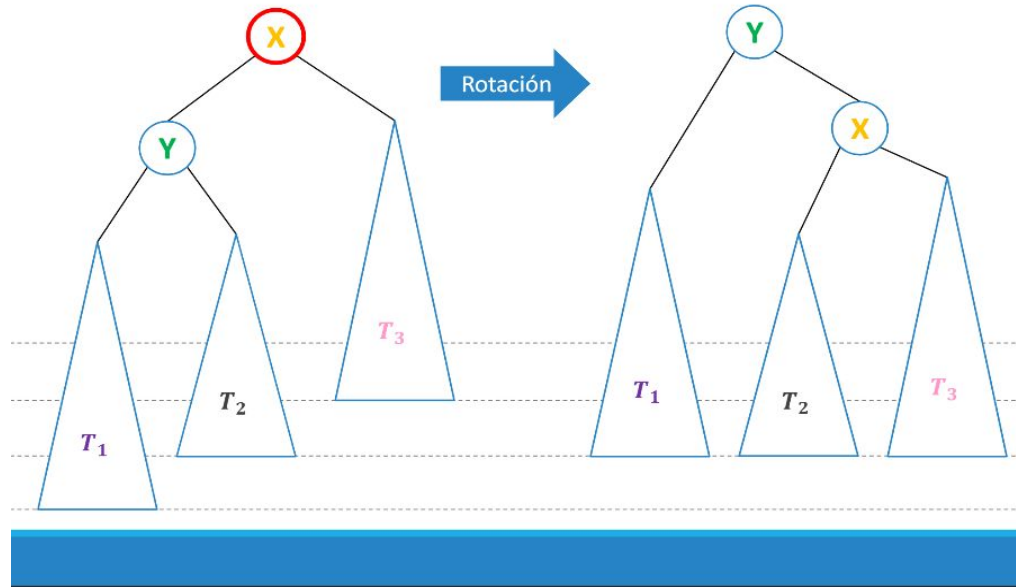
www.penjee.com

Problema 1

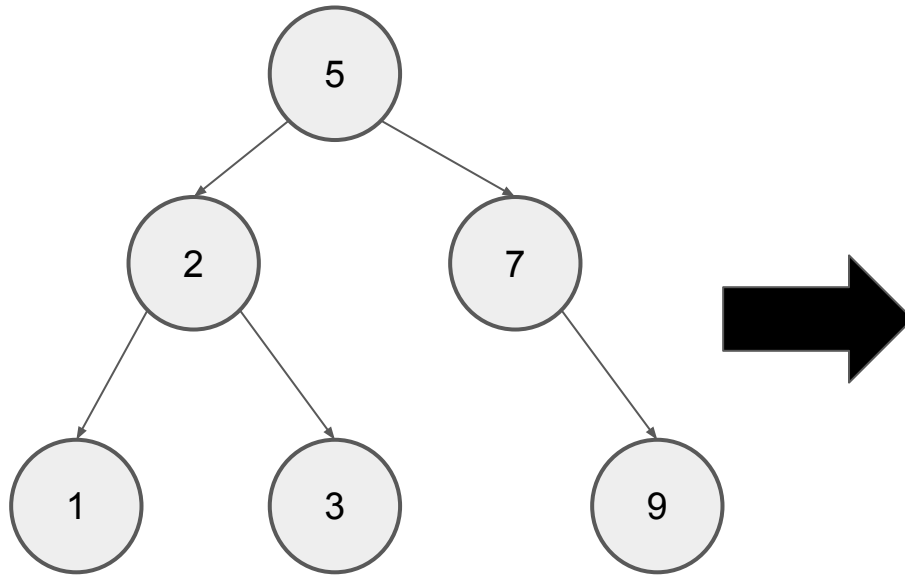
- Encuentre una forma de pasar de un árbol **BST** a otro árbol **BST** (que tengan los mismo elementos), por medio de rotaciones en $O(n)$.
- Encuentre una forma de convertir un árbol BST a un array ordenado en $O(n)$.

Recordemos la rotación

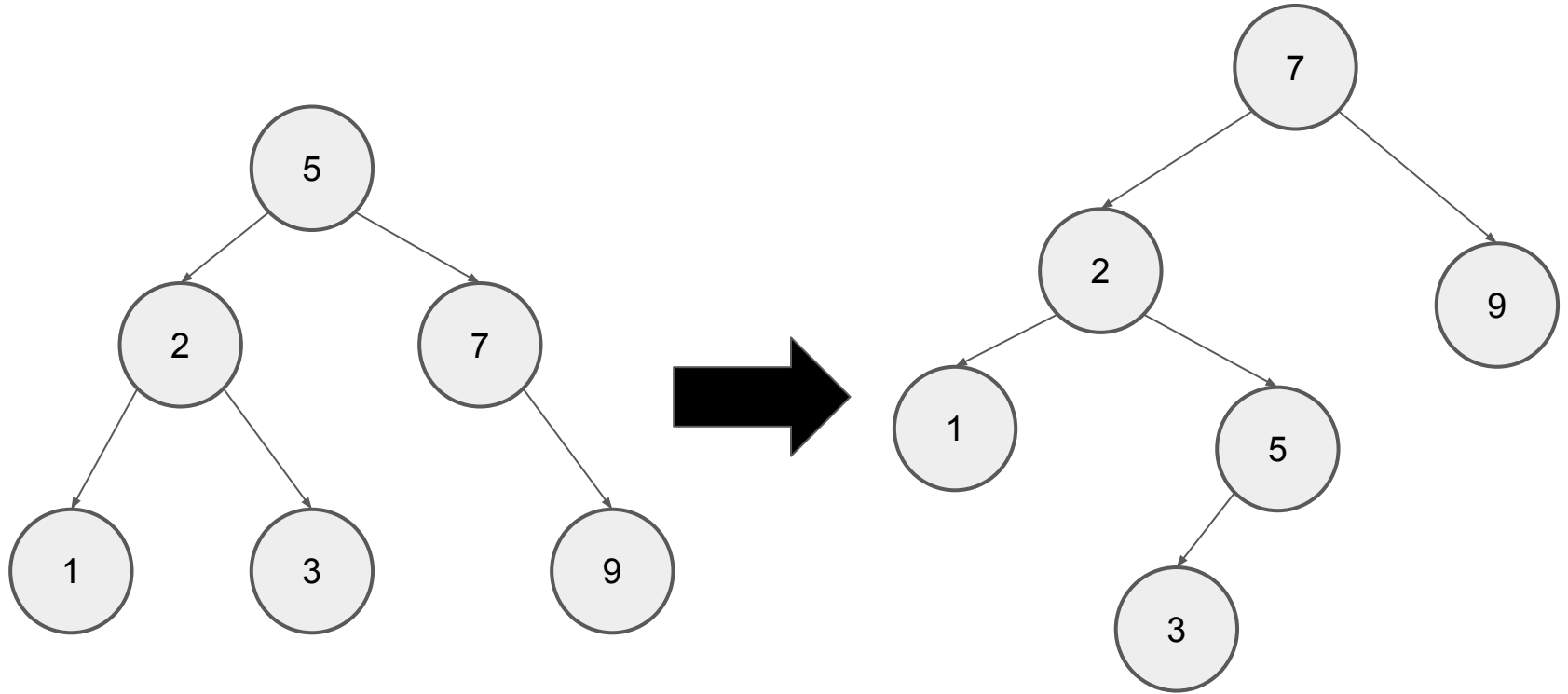
Rotación a la derecha en torno a X-Y



Problema 1



Problema 1

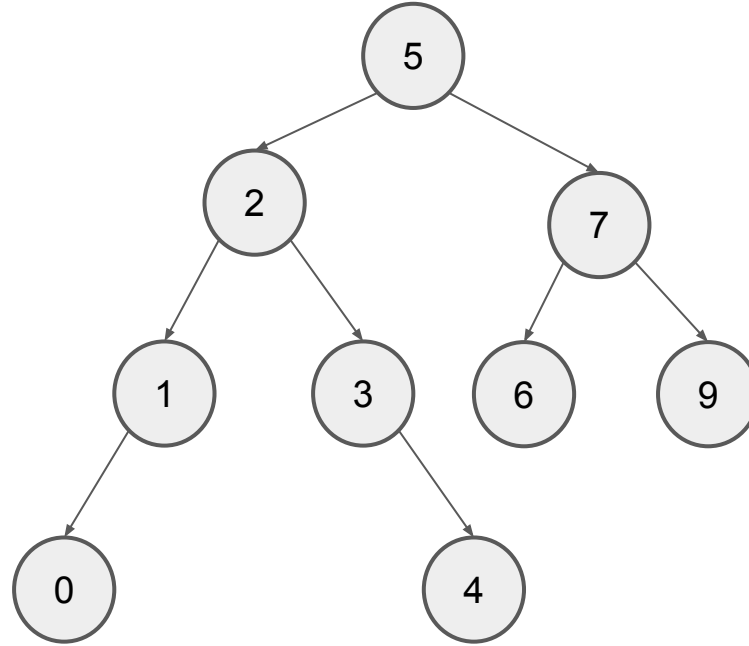


Problema 1

- Encuentre una forma de pasar de un árbol BST a otro árbol BST (que tengan los mismo elementos), por medio de rotaciones en $O(n)$.
- Encuentre una forma de convertir un árbol BST a un array ordenado en $O(n)$.

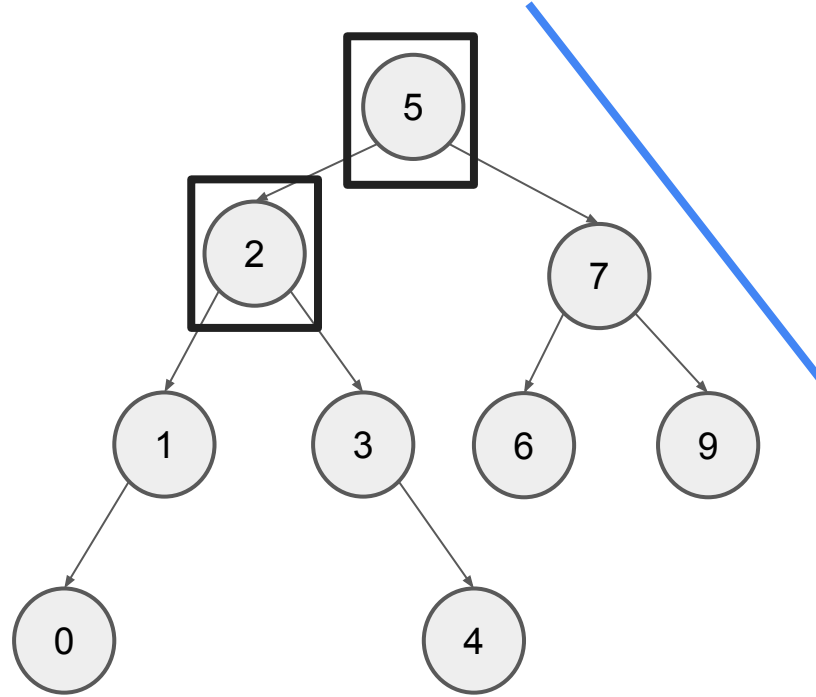
¿Por qué resolver el primer problema me permite resolver el otro?

Problema 1

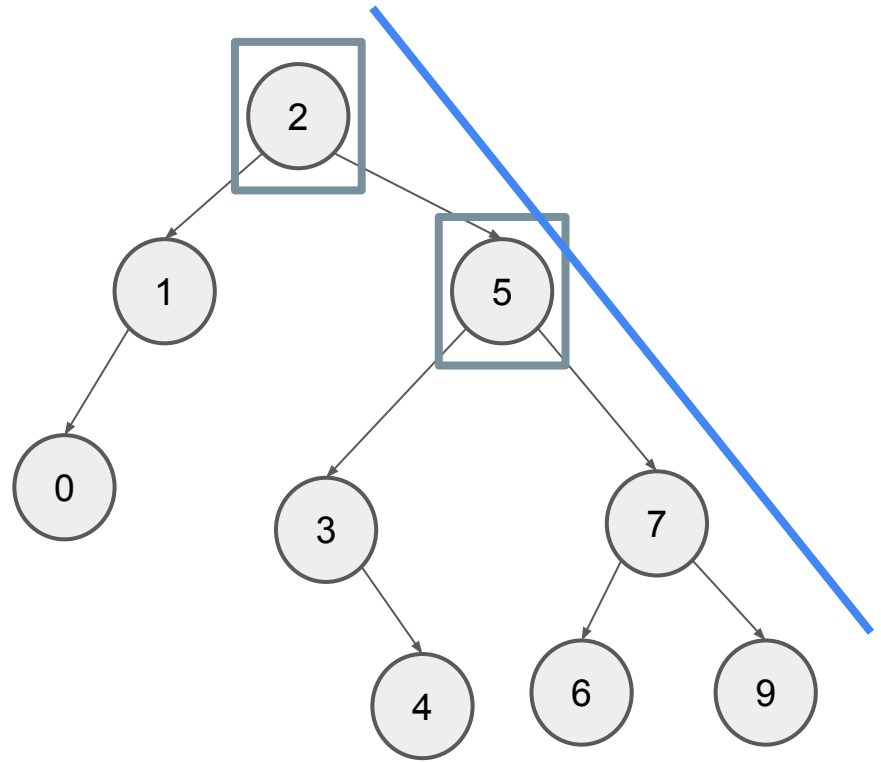


Problema 1

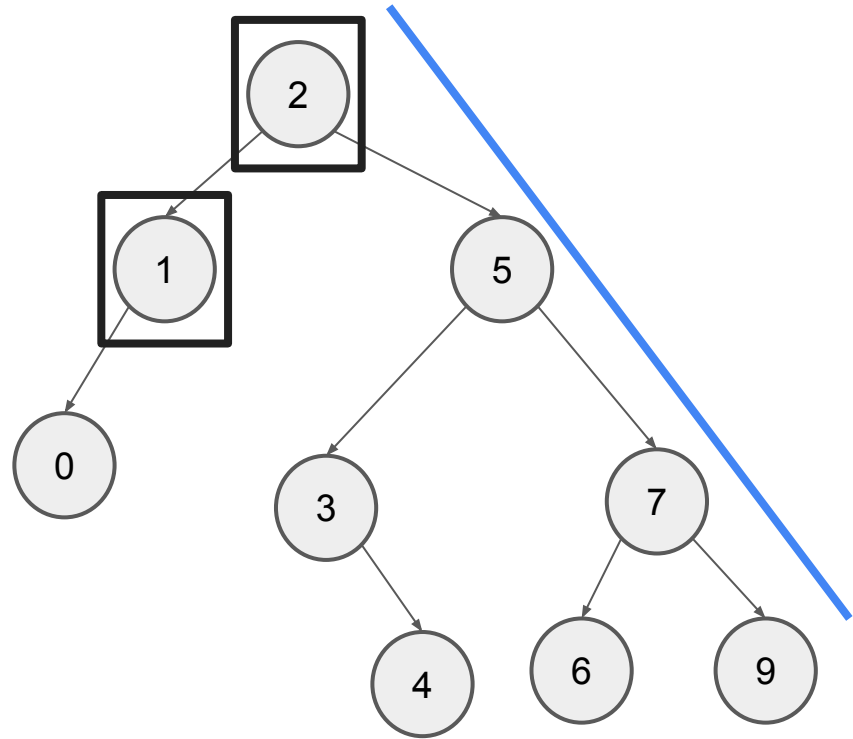
Vamos a hacer rotaciones siempre en una misma dirección desde el lugar más alto dentro del árbol (donde sea posible hacer una rotación)



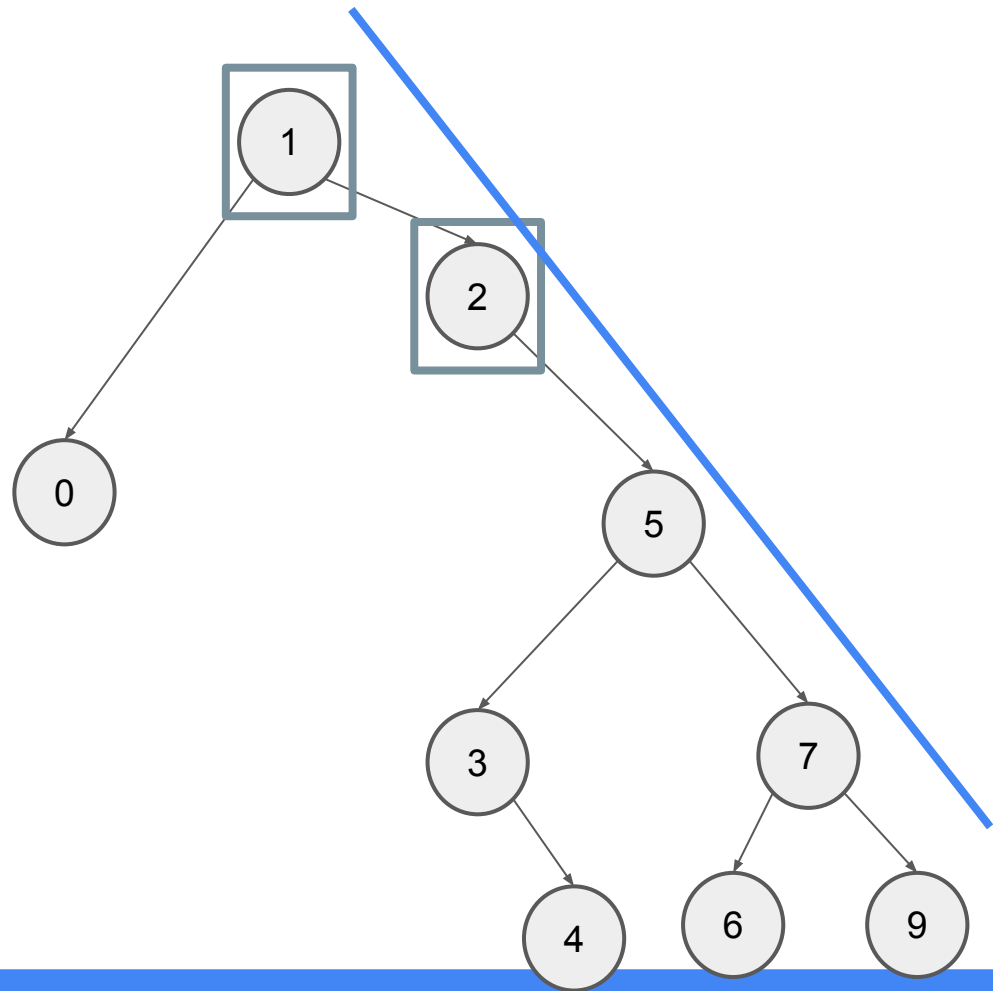
Problema 1



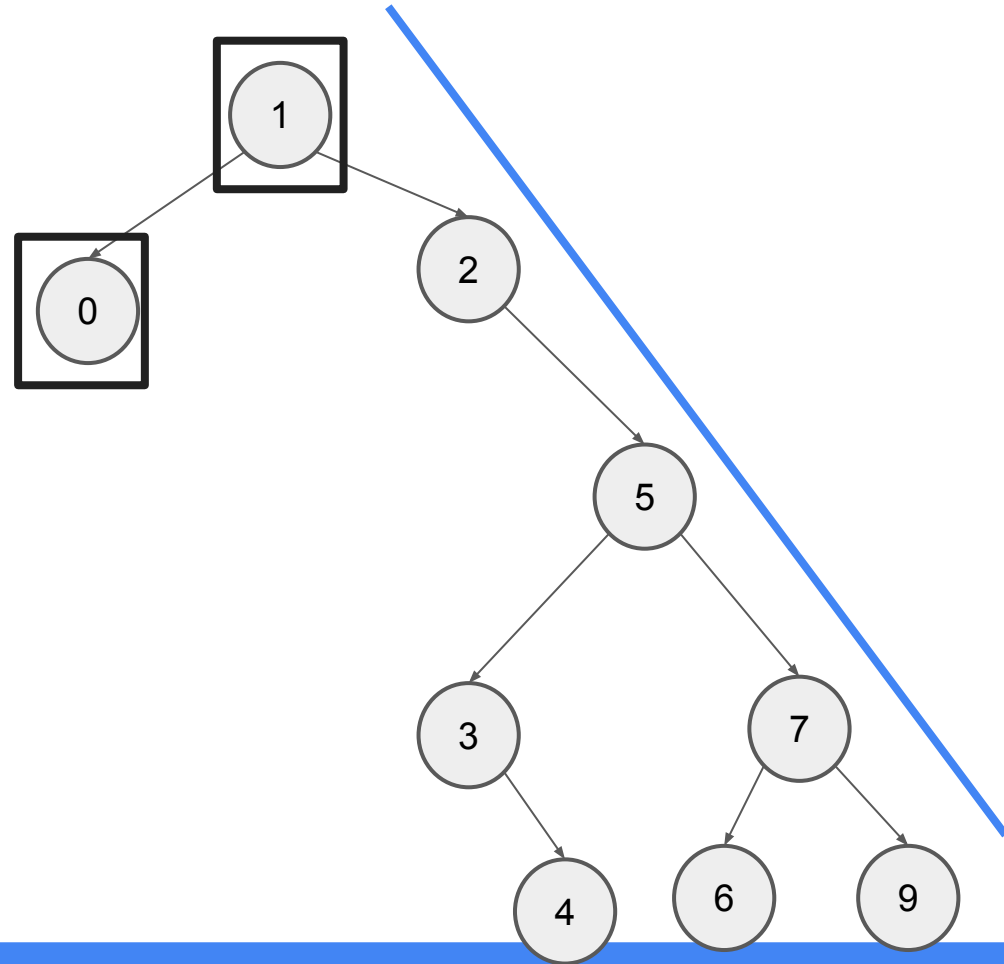
Problema 1



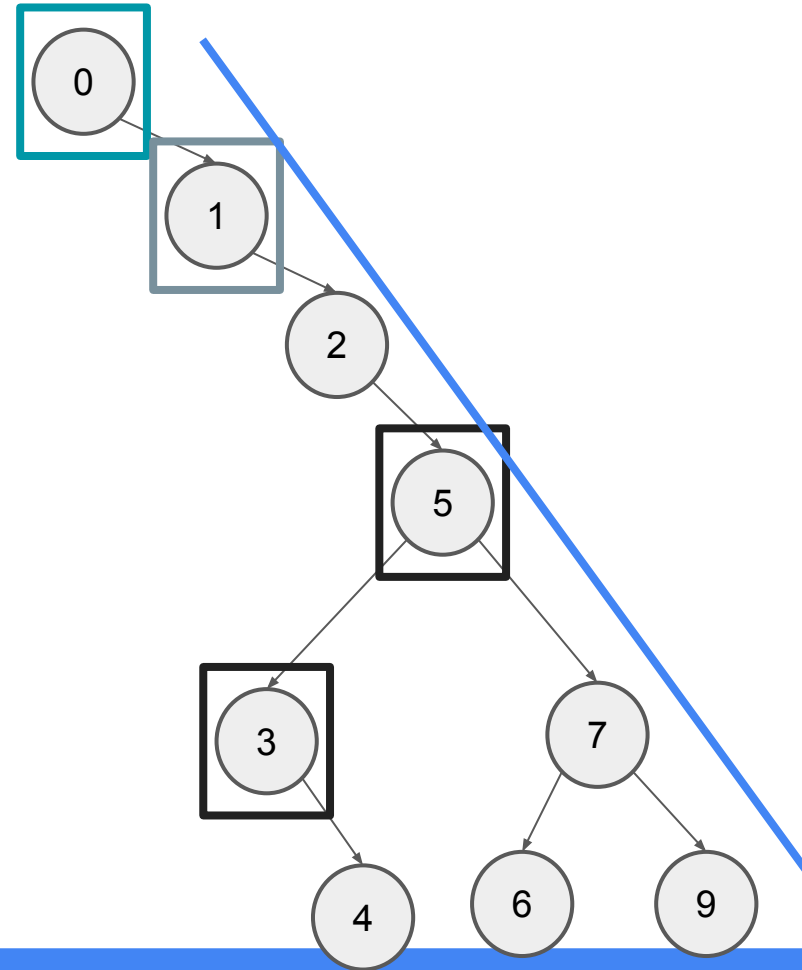
Problema 1



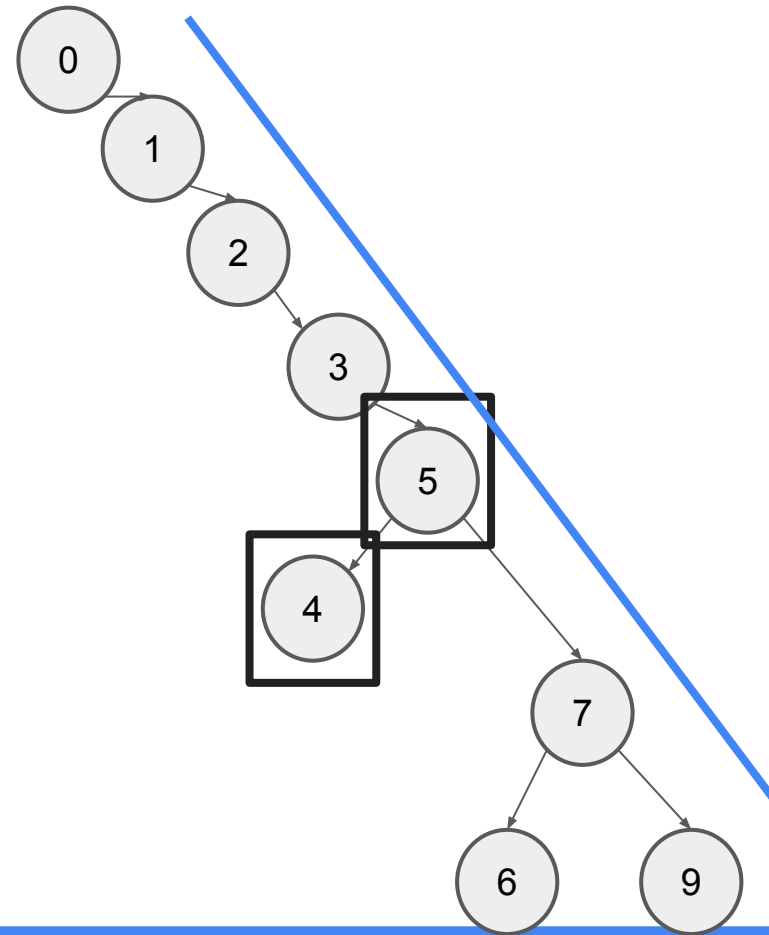
Problema 1



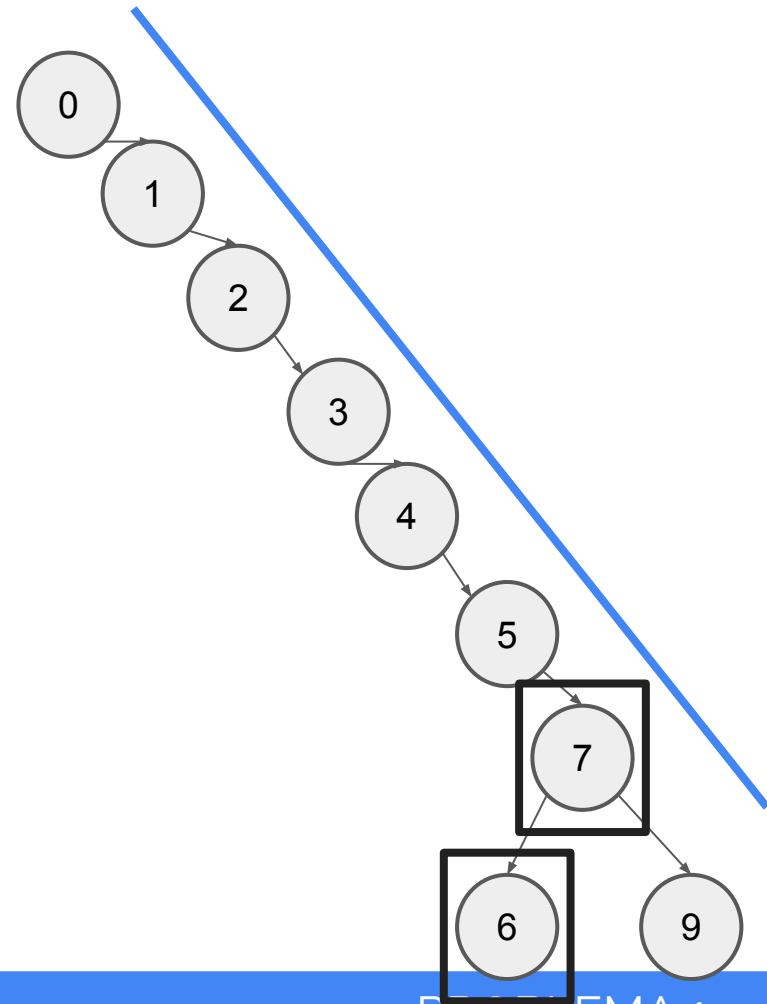
Problema 1



Problema 1



Problema 1



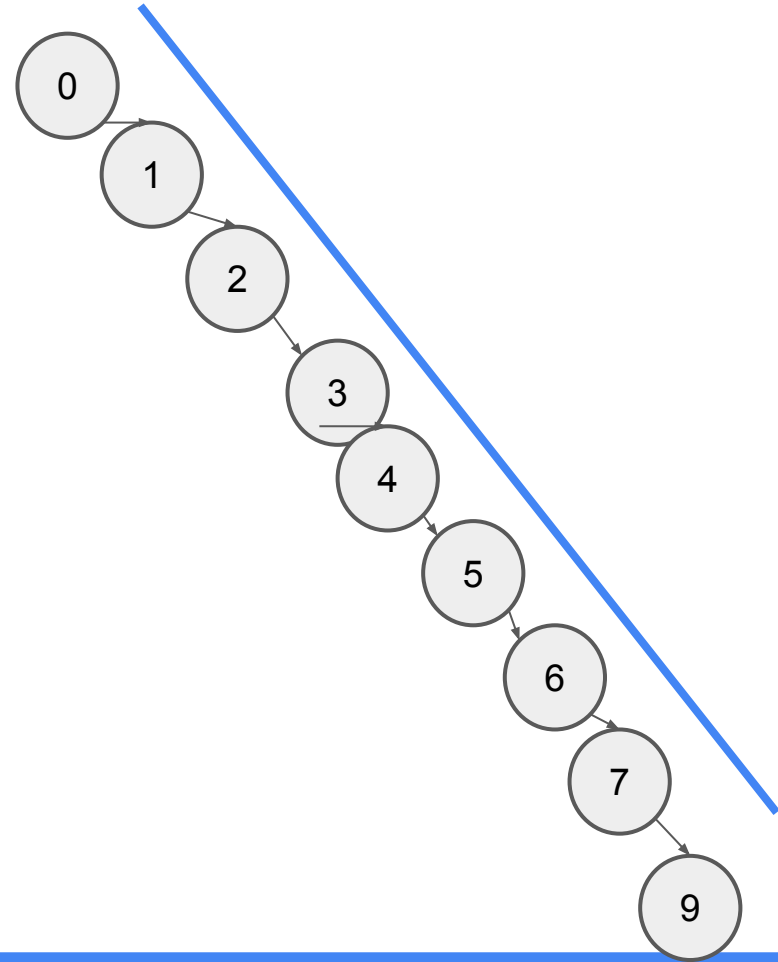
Problema 1

Lo logramos!

¿Por qué funciona?

¿Por qué funciona en $O(n)$?

¿Como nos permite concluir el problema?



Problema 2

Muestre la secuencia de árbol **AVL** que se forma al insertar las claves 3, 2, 1, 4, 5, 6, 7, 16, 15 y 14, en este orden, en un árbol AVL inicialmente vacío.

Problema 2



[3, 2, 1, 4, 5, 6, 7, 16, 15, 14]

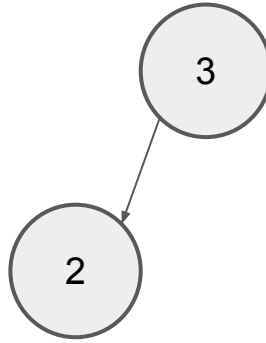
Problema 2



[3, 2, 1, 4, 5, 6, 7, 16, 15, 14]

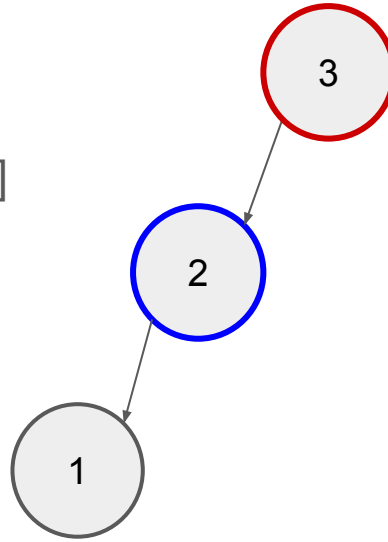
Problema 2

[3, 2, 1, 4, 5, 6, 7, 16, 15, 14]



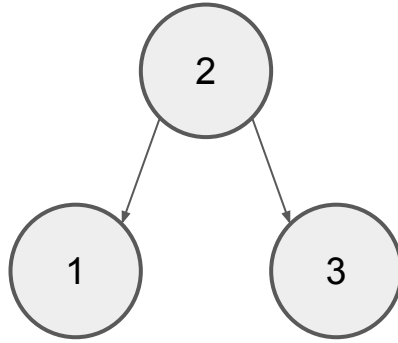
Problema 2

[3, 2, 1, 4, 5, 6, 7, 16, 15, 14]



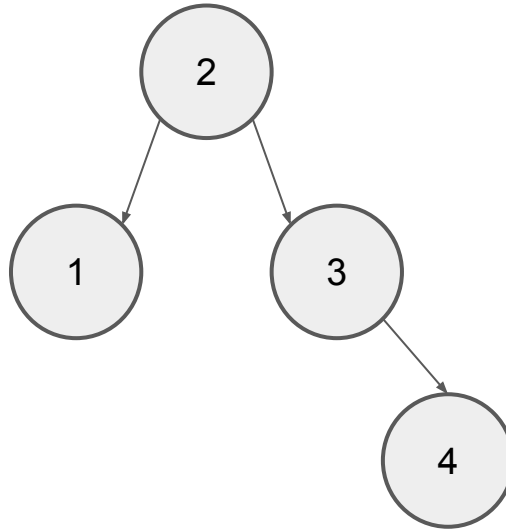
Problema 2

[3, 2, 1, 4, 5, 6, 7, 16, 15, 14]



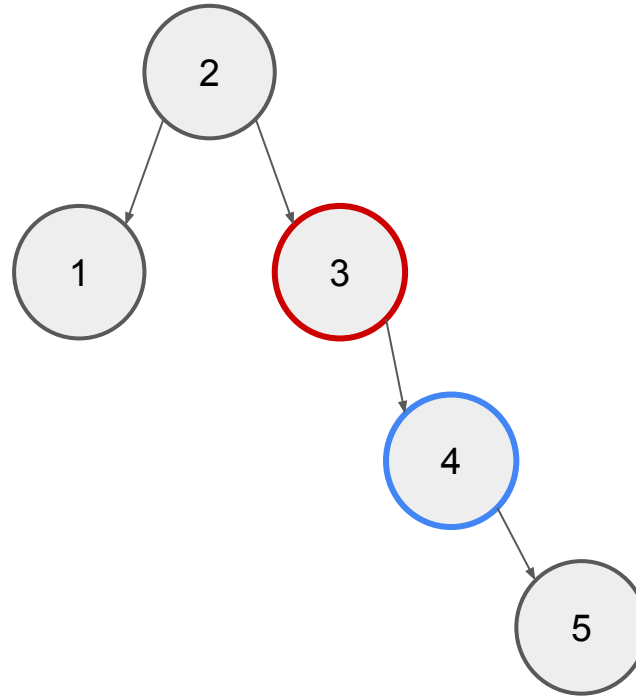
Problema 2

[3, 2, 1, 4, 5, 6, 7, 16, 15, 14]



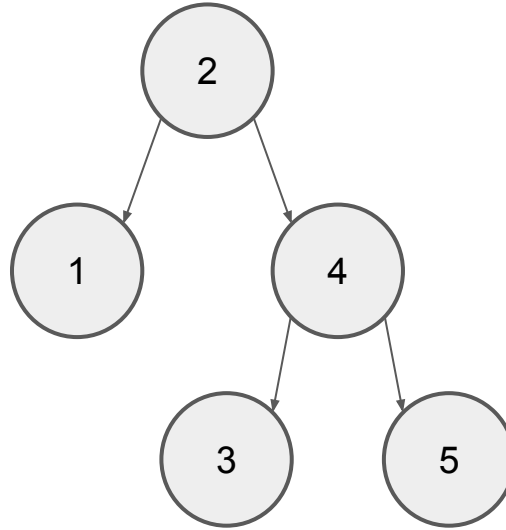
Problema 2

[3, 2, 1, 4, 5, 6, 7, 16, 15, 14]



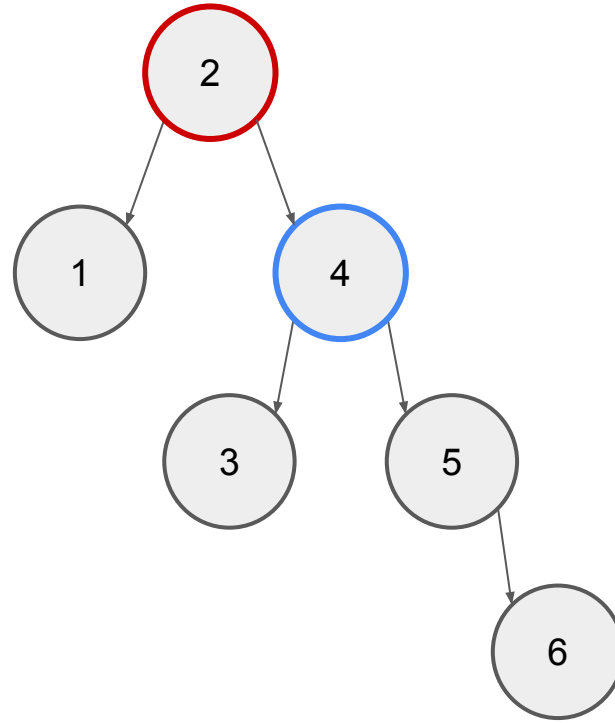
Problema 2

[3, 2, 1, 4, 5, 6, 7, 16, 15, 14]



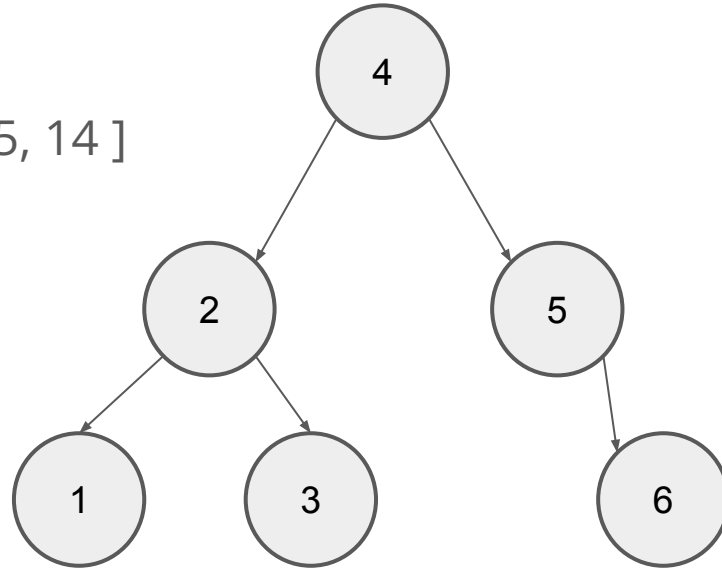
Problema 2

[3, 2, 1, 4, 5, 6, 7, 16, 15, 14]



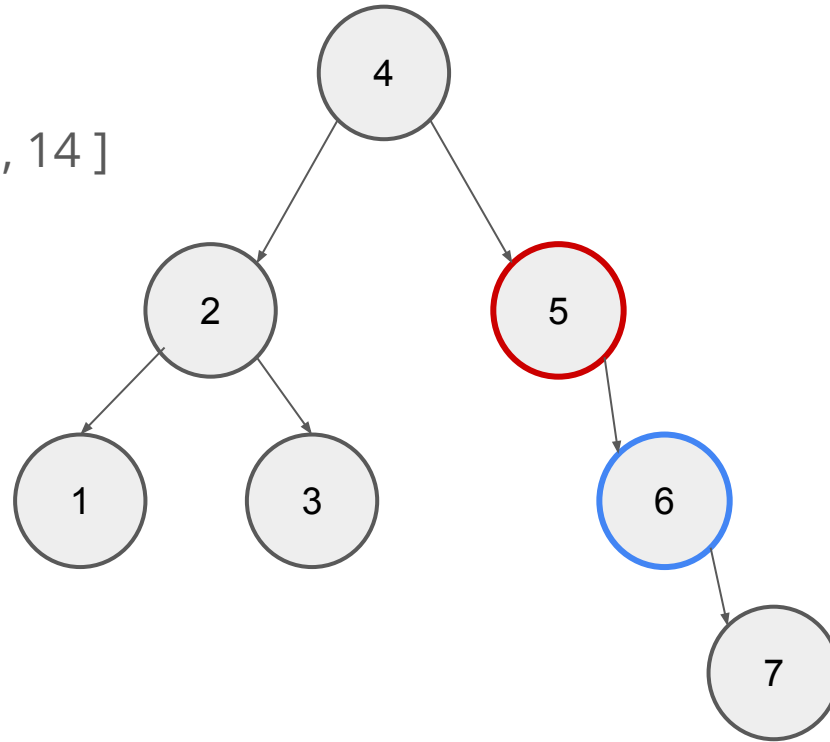
Problema 2

[3, 2, 1, 4, 5, 6, 7, 16, 15, 14]



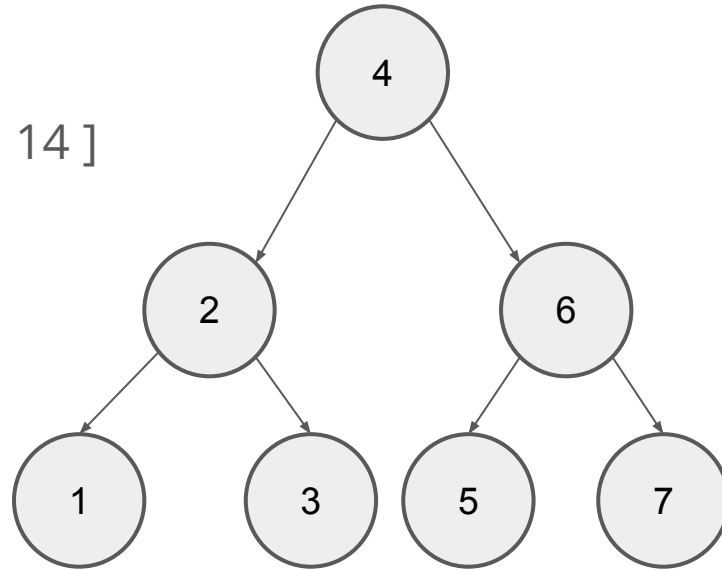
Problema 2

[3, 2, 1, 4, 5, 6, 7, 16, 15, 14]



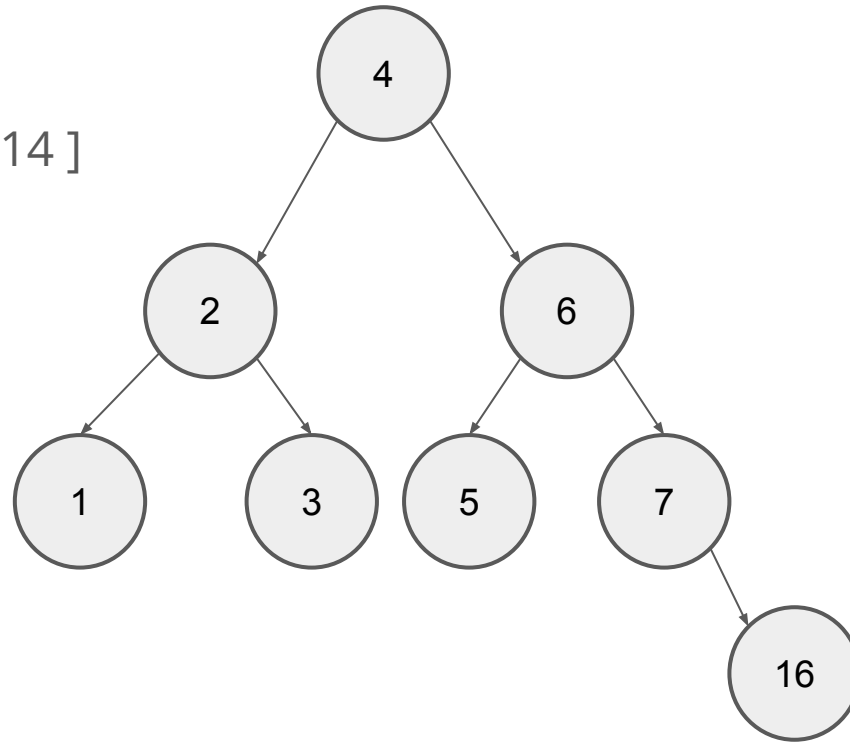
Problema 2

[3, 2, 1, 4, 5, 6, 7, 16, 15, 14]



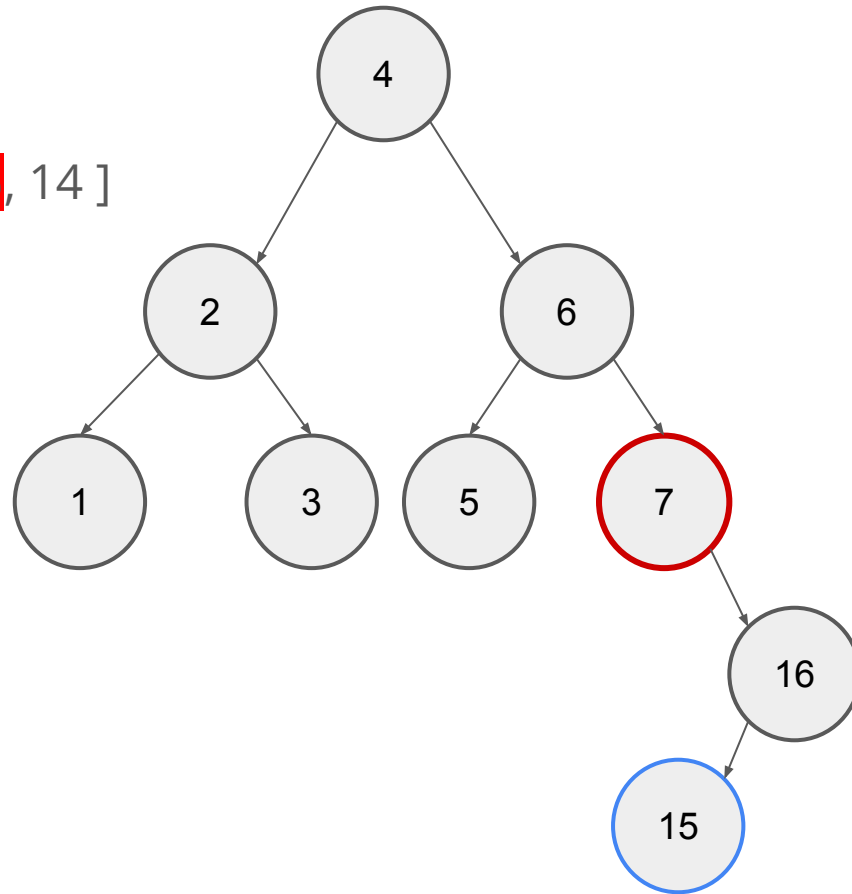
Problema 2

[3, 2, 1, 4, 5, 6, 7, 16, 15, 14]



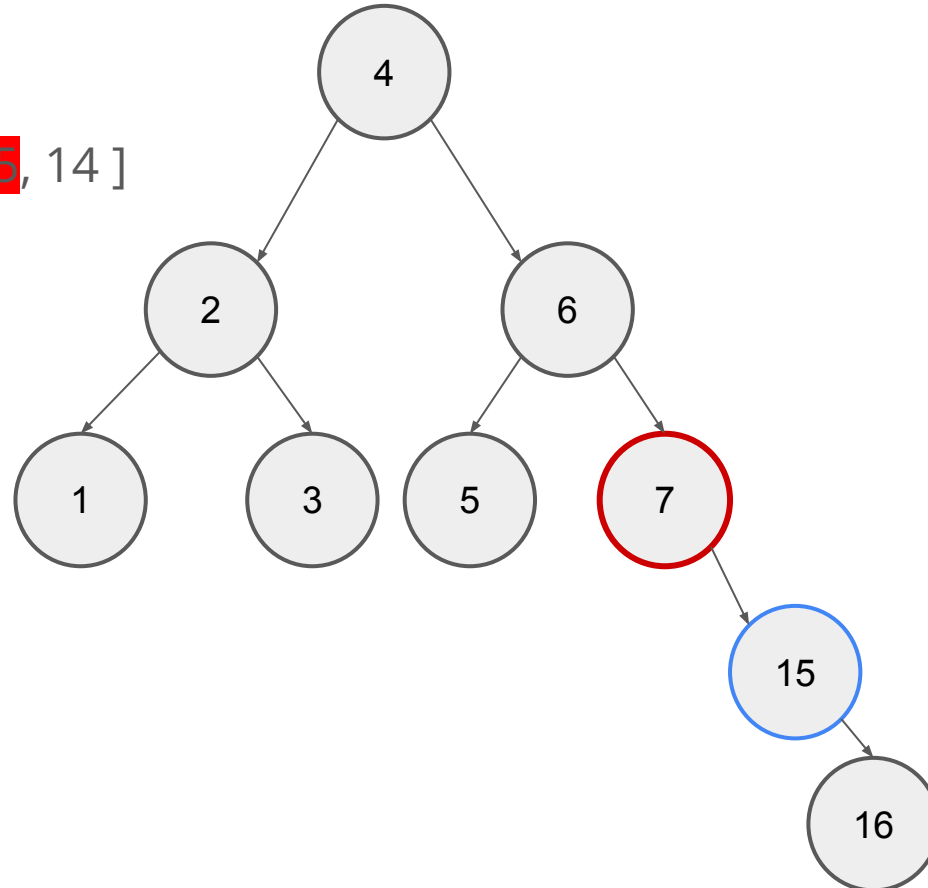
Problema 2

[3, 2, 1, 4, 5, 6, 7, 16, 15, 14]



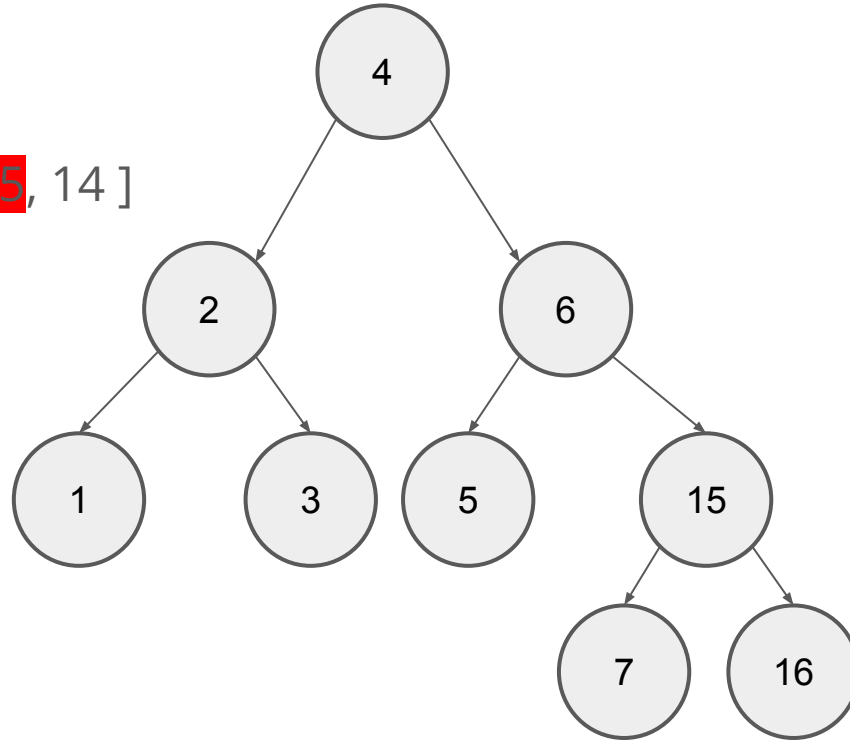
Problema 2

[3, 2, 1, 4, 5, 6, 7, 16, 15, 14]



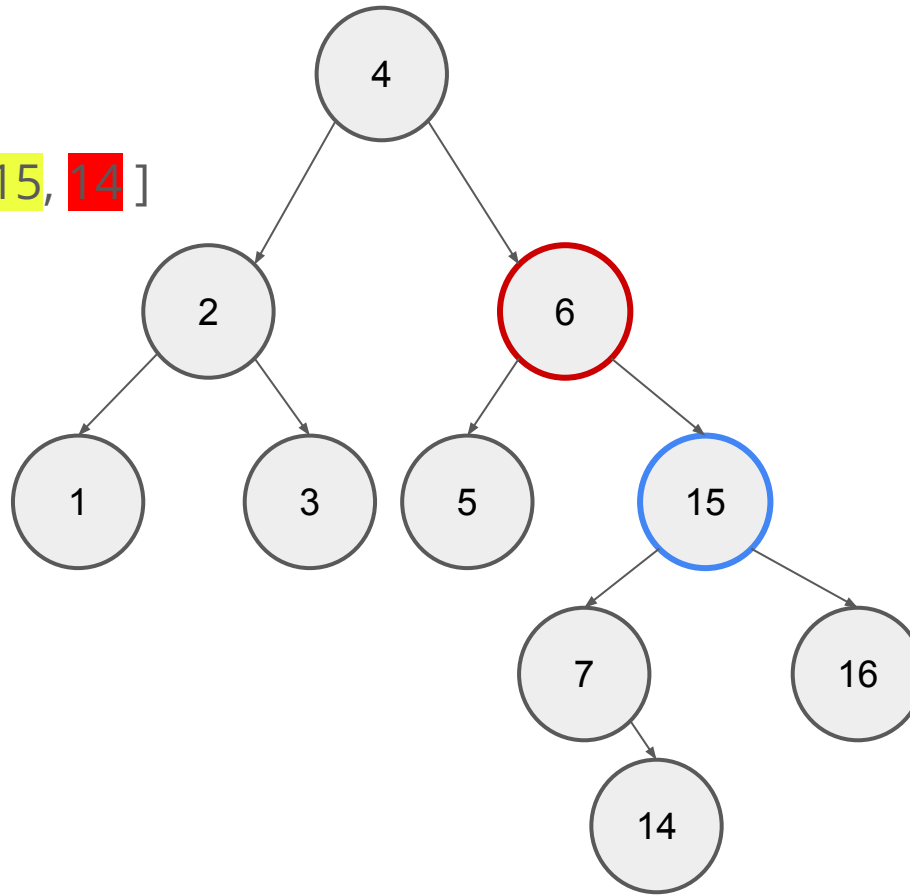
Problema 2

[3, 2, 1, 4, 5, 6, 7, 16, 15, 14]



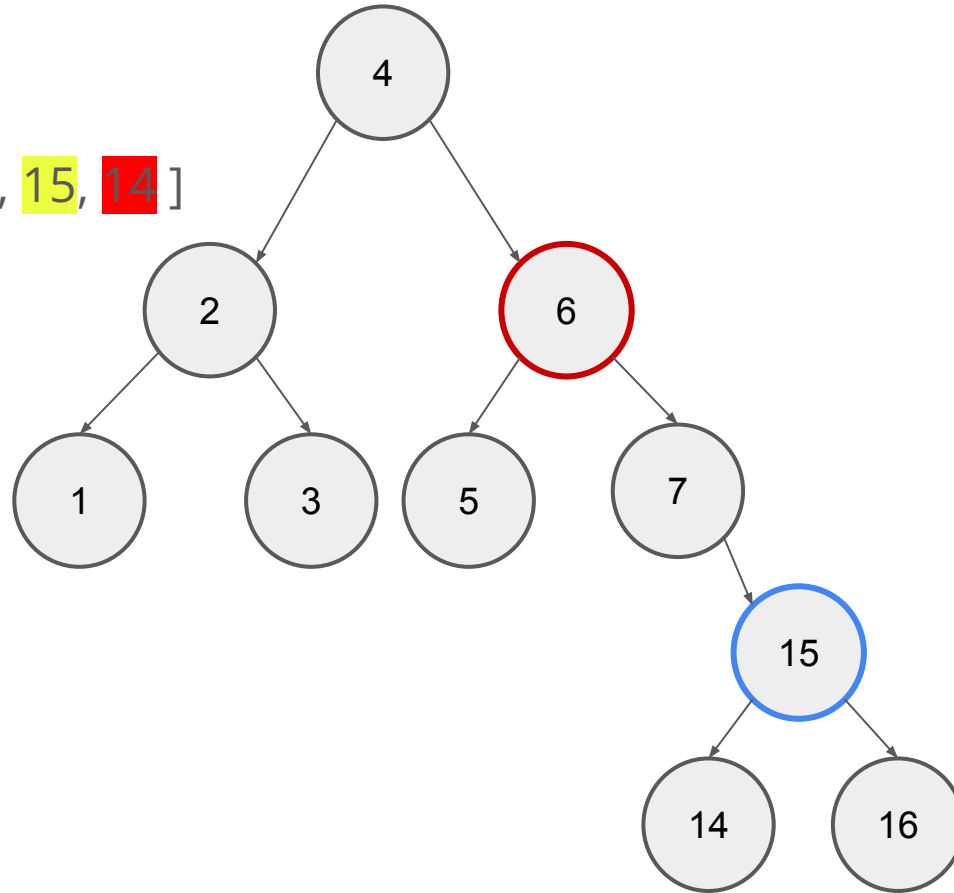
Problema 2

[3, 2, 1, 4, 5, 6, 7, 16, 15, 14]



Problema 2

[3, 2, 1, 4, 5, 6, 7, 16, 15, 14]



Problema 2

[3, 2, 1, 4, 5, 6, 7, 16, 15, 14]

