

ÁRBOLES

Árboles

Contenidos a revisar

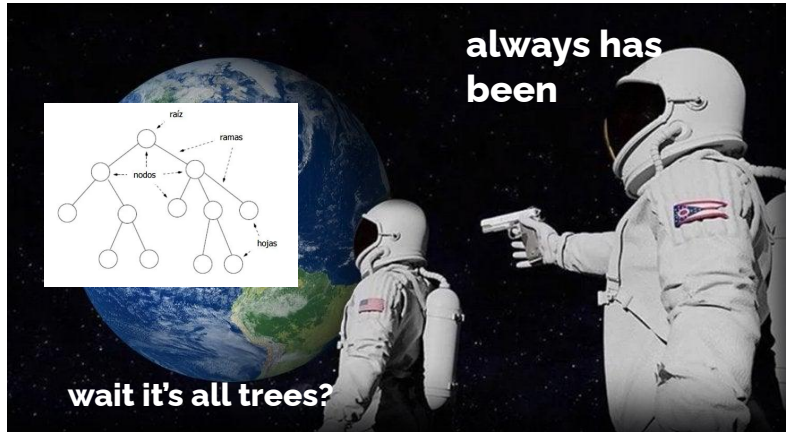
ABB

-

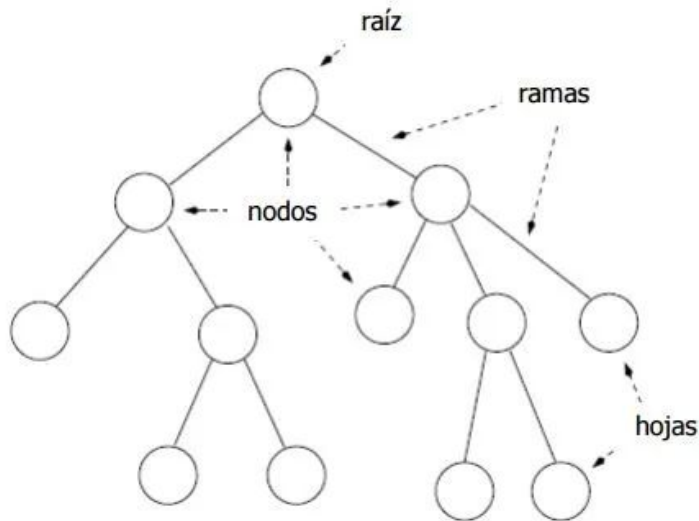
AVL

-

Rojo-Negro

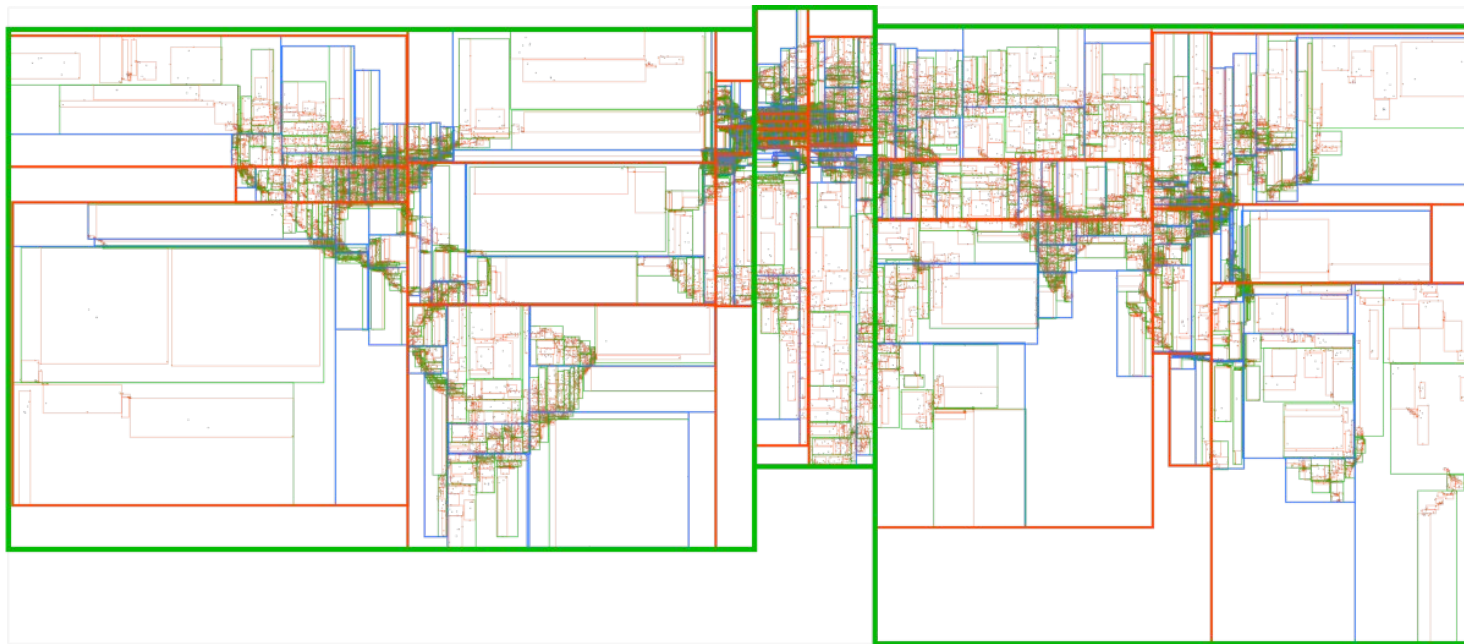


Que es un árbol



- Estructura de datos con nodos
- Tiene una raíz
- Cada nodo puede tener 0 o más hijos
- Cada nodo tiene 1 padre (menos el nodo raíz)
- No hay ciclos
- Se arman con normas pre-establecidas

Un ejemplo Gráfico, KD-Tree



Árbol Binario de Búsqueda (ABB)

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

Árbol AVL

Propiedades:

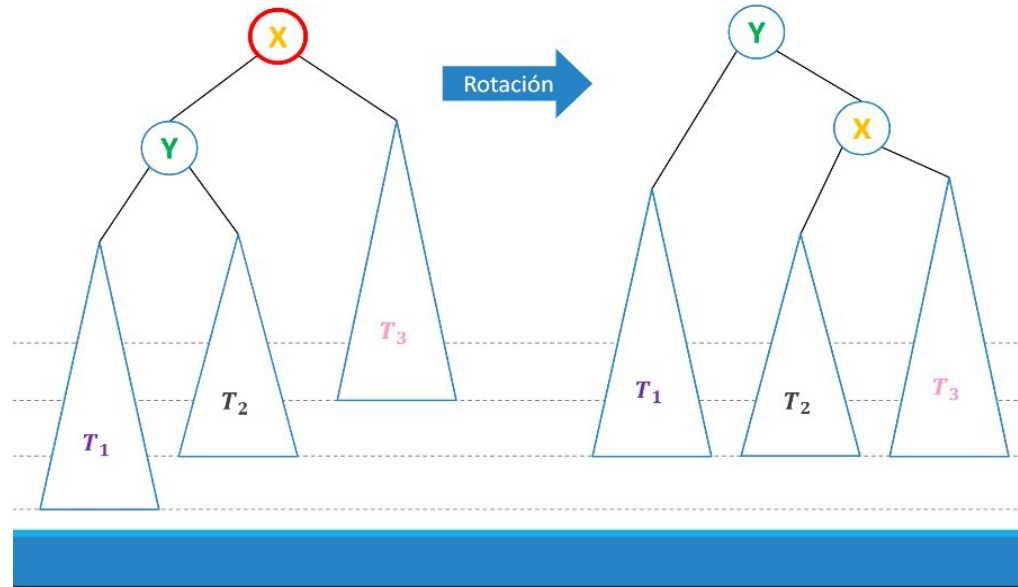
- Es un árbol binario de búsqueda (ABB)
- Está balanceado
- Las alturas de los hijos de la raíz difieren a lo más de 1 entre ellos
- Cada hijo es AVL
- Altura es $\log(n)$

Problema 1

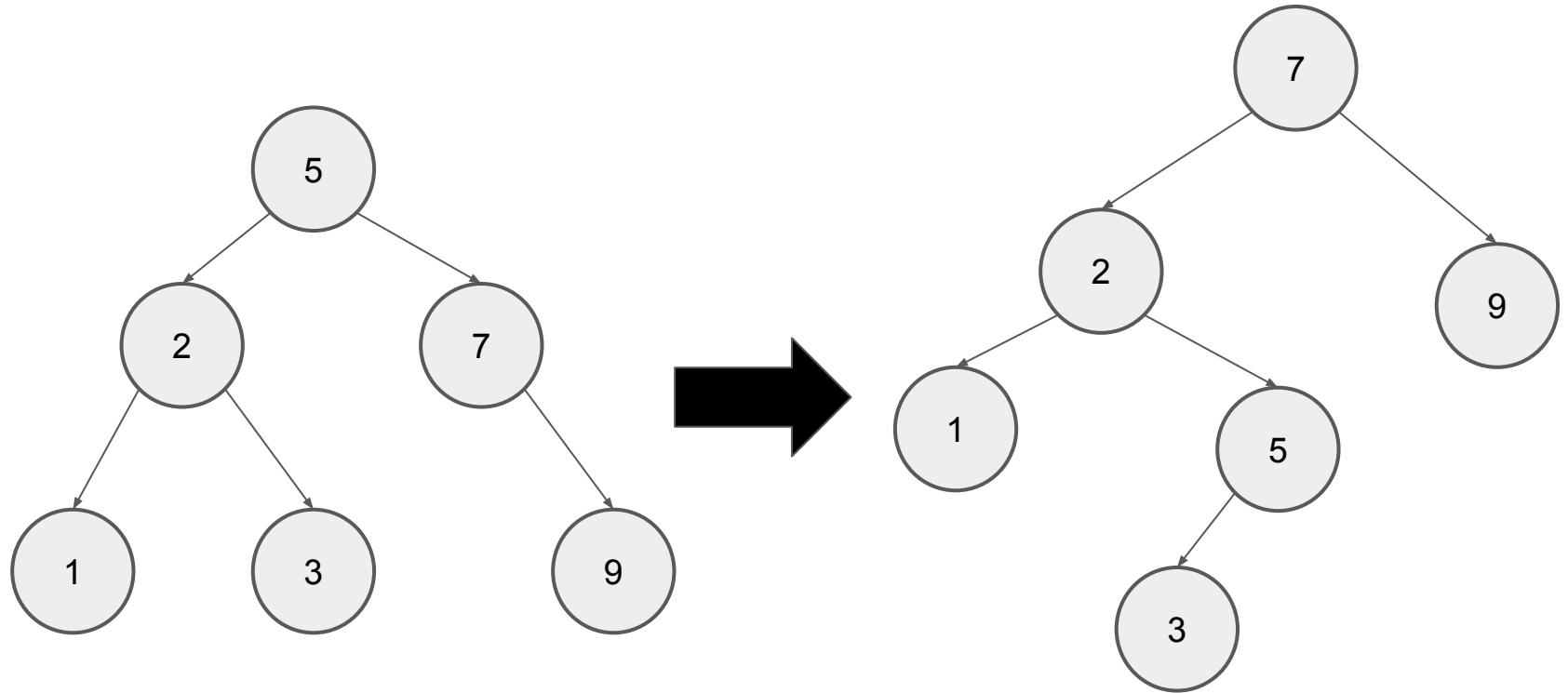
- Encuentre una forma de pasar de un árbol ABB a otro árbol ABB (que tengan los mismo elementos), por medio de rotaciones en $O(n)$.
- Encuentre una forma de convertir un árbol ABB a un array ordenado en $O(n)$.

Recordemos la rotación

Rotación a la derecha en torno a X-Y



Problema 1

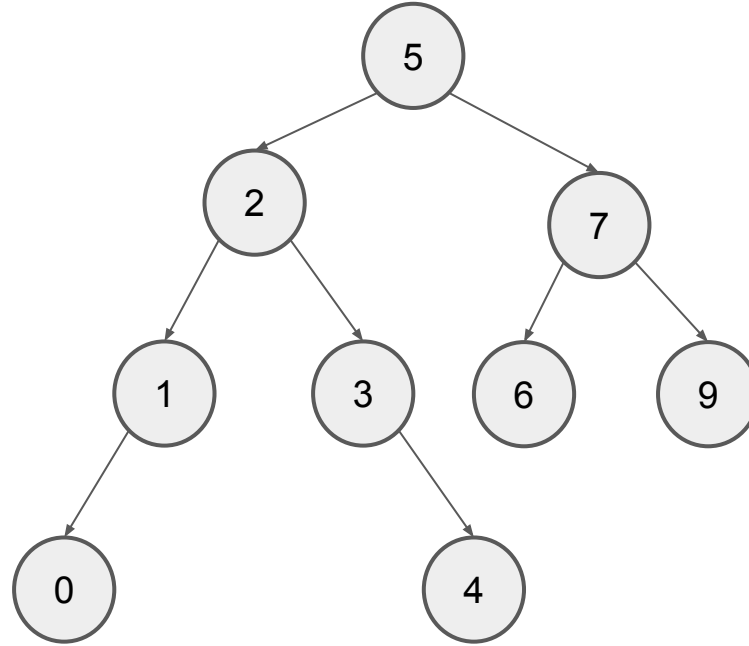


Problema 1

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

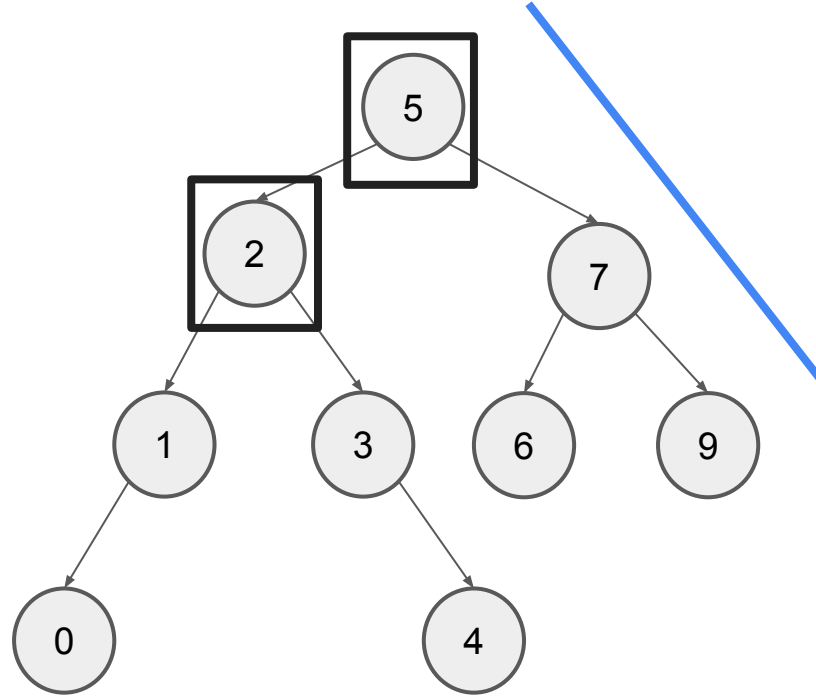
¿Por qué resolver el primer problema me permite resolver el otro?

Problema 1

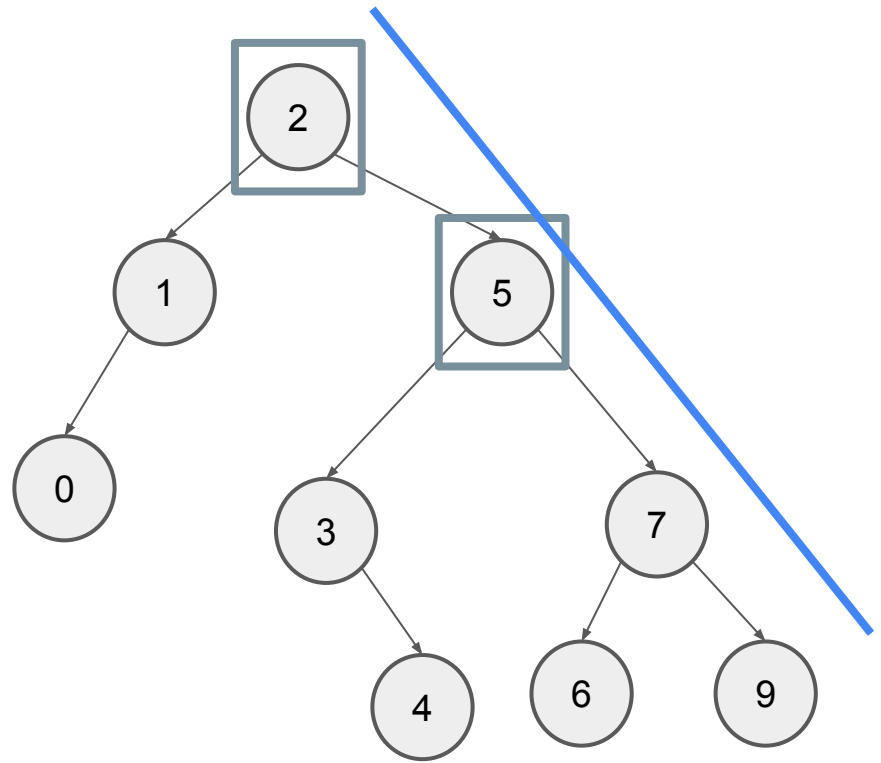


Problema 1

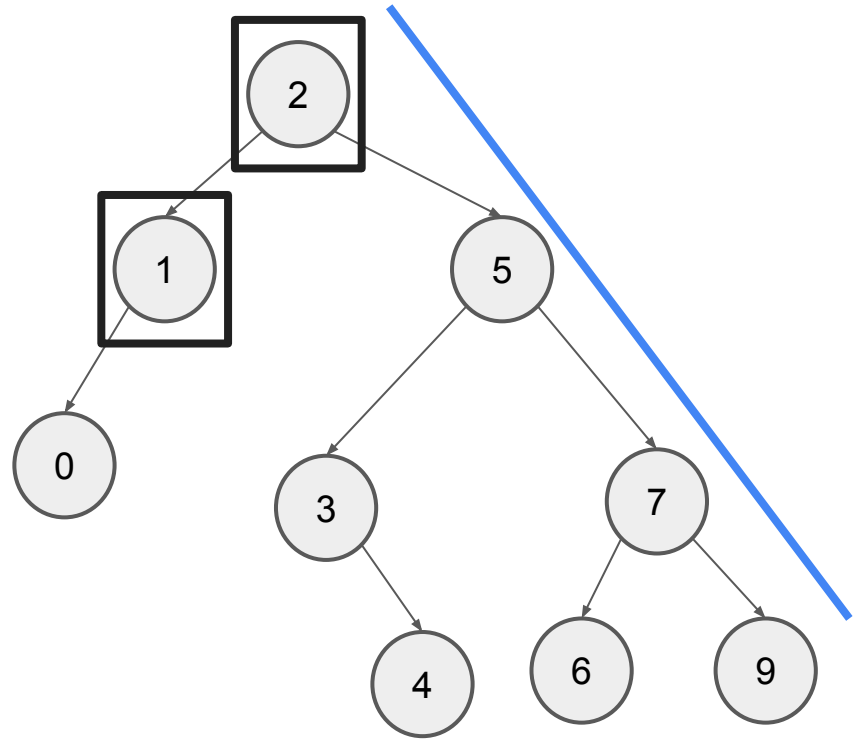
Vamos a hacer rotaciones siempre en una misma dirección desde el lugar más alto dentro del árbol (donde sea posible hacer una rotación)



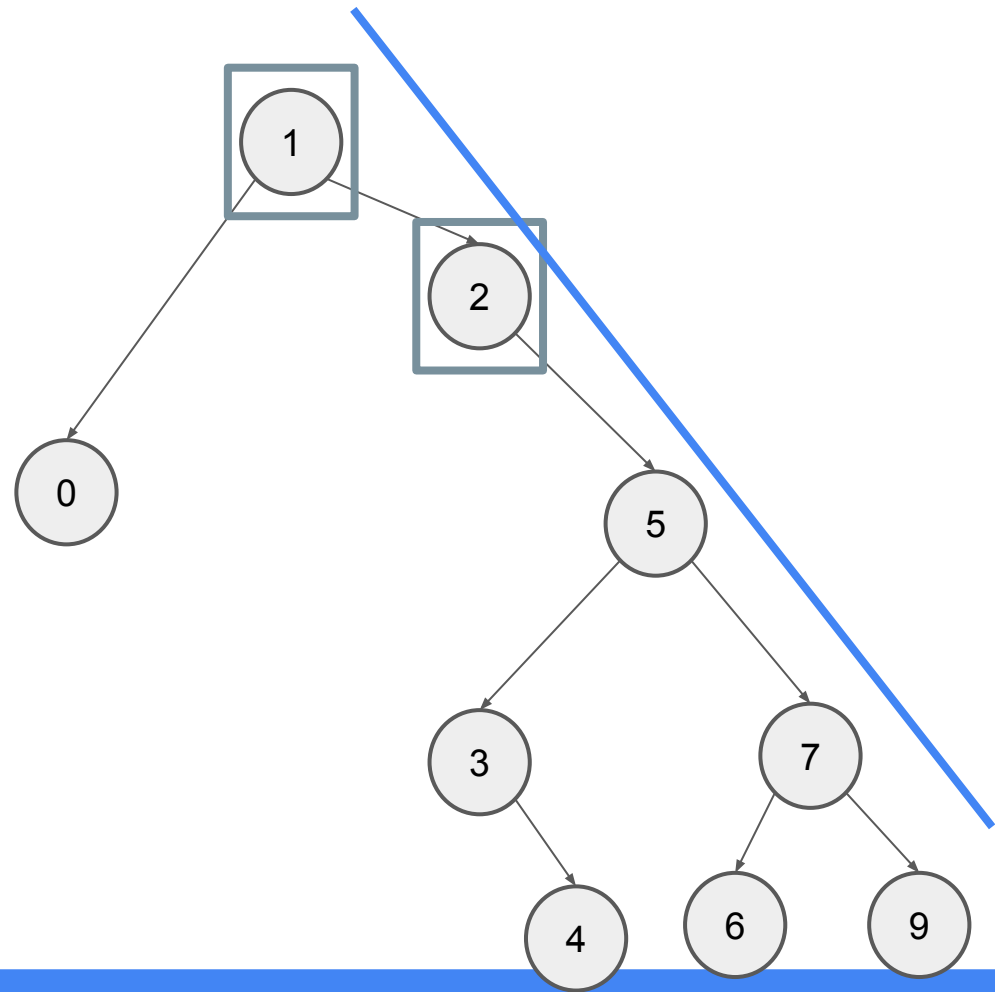
Problema 1



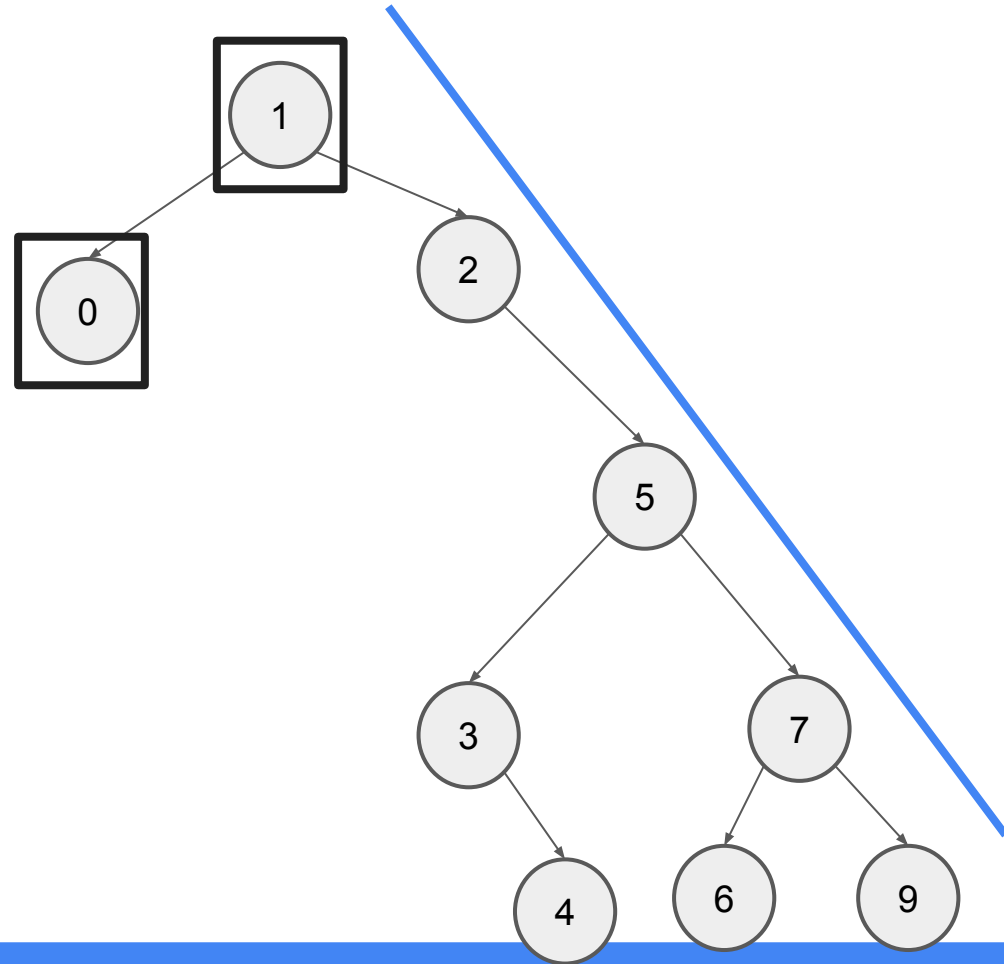
Problema 1



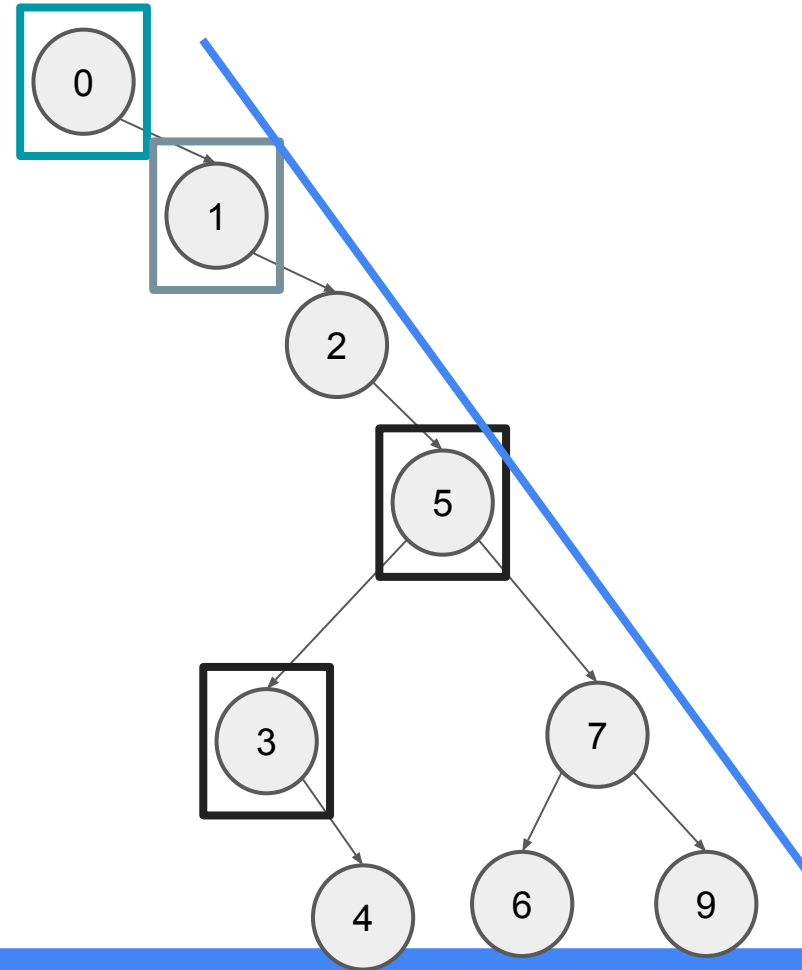
Problema 1



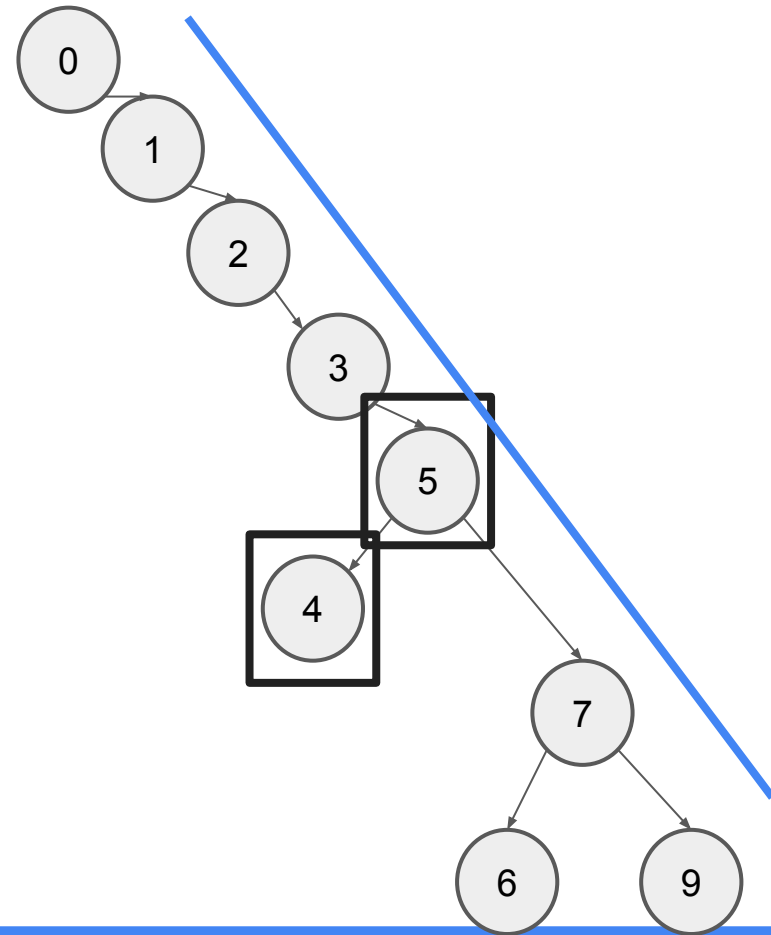
Problema 1



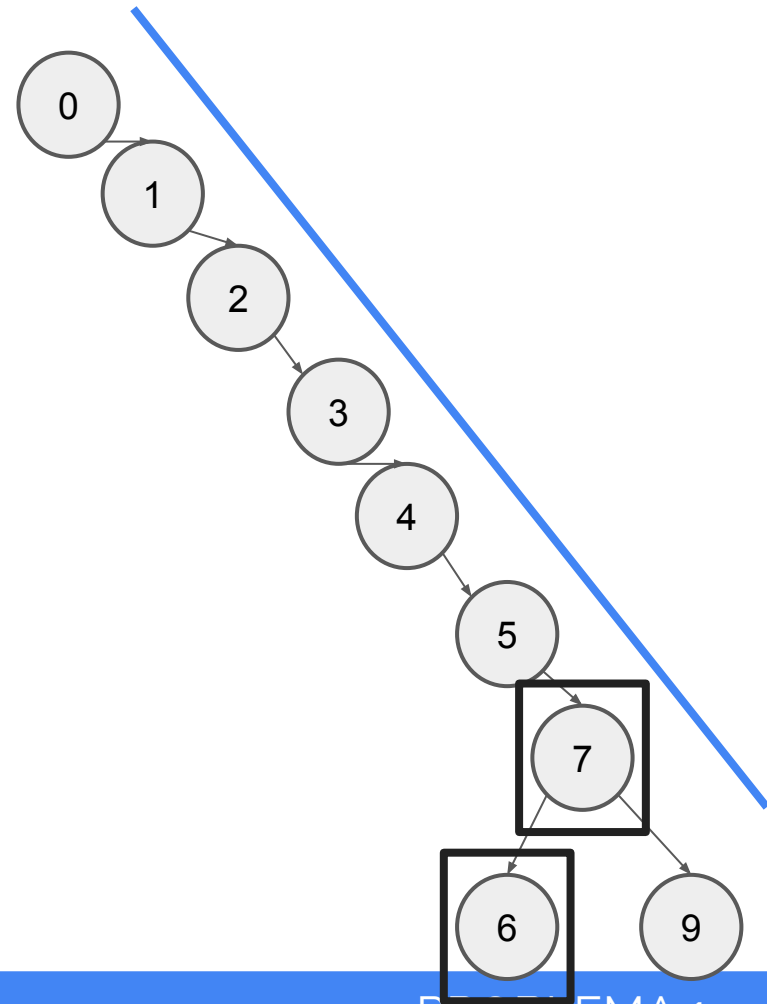
Problema 1



Problema 1



Problema 1



Problema 1

Lo logramos!

¿Por qué funciona?

¿Por qué funciona en $O(n)$?

¿Como nos permite concluir el problema?

