

# ÁRBOLES BINARIOS EQUILIBRADOS

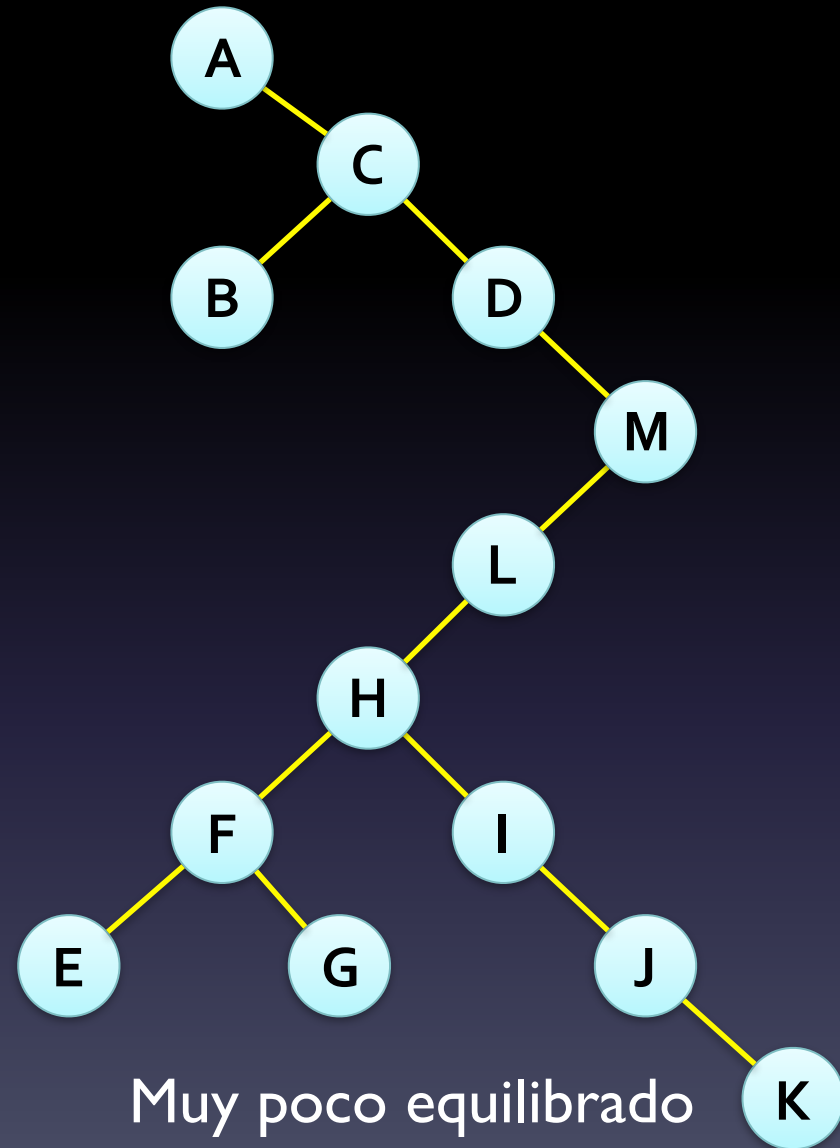
## AVL

# Motivación

- En ocasiones, la construcción de los ABB conduce a árboles con características muy pobres para la búsqueda
- Ejemplo: Construir un ABB con {A, C, D, M, L, H, I, B, F, G, J, K, E}

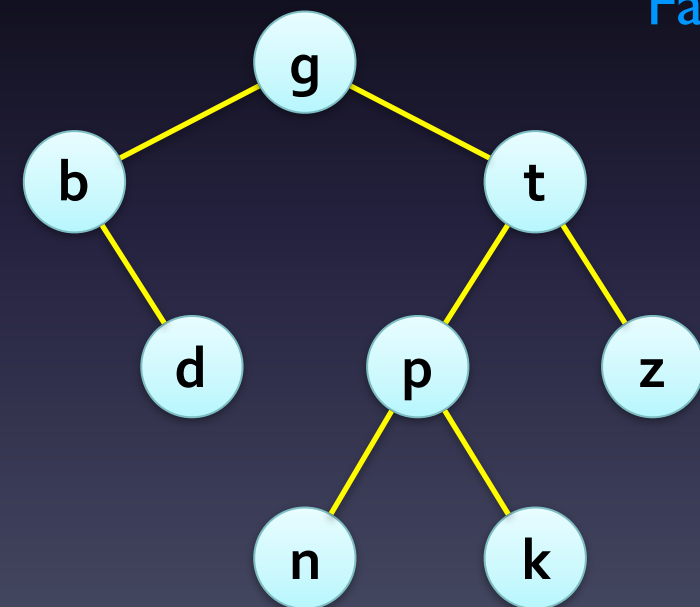
## IDEA

Construir ABB equilibrados, impidiendo que en ningún nodo las alturas de los subárboles izquierdo y derecho difieran en más de una unidad



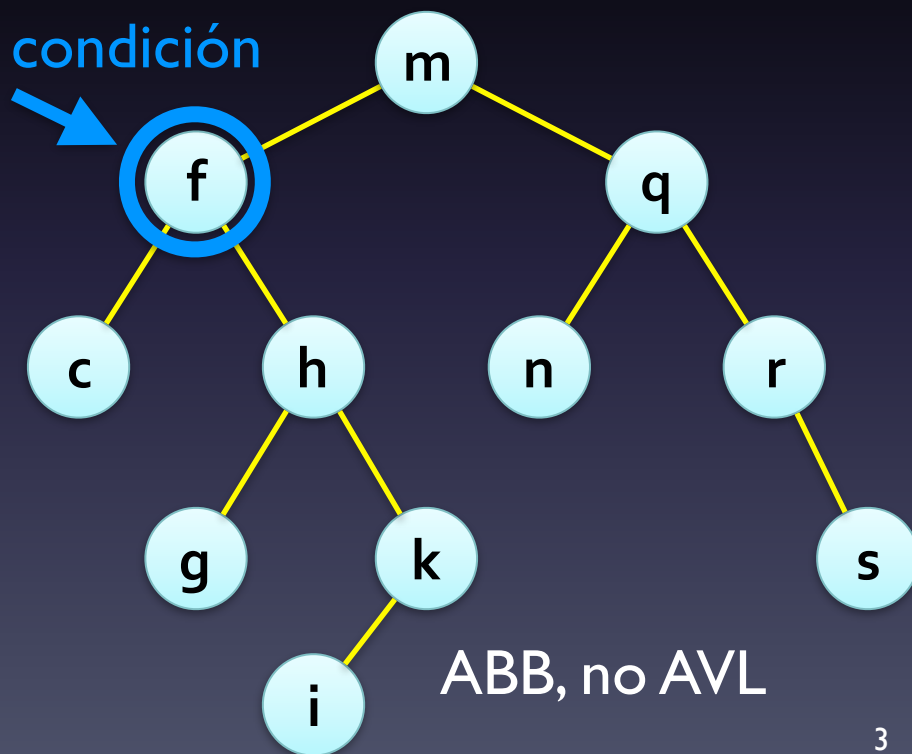
# Árboles AVL

- Diremos que un árbol binario ABB es un AVL (o que está equilibrado en el sentido de Addelson-Velski-Landis) si para cada uno de sus nodos se cumple que las alturas de sus dos subárboles difieren como máximo en 1
- Ejemplo:



AVL (ABB + Equilibrio)

Falla la condición



ABB, no AVL

# Árboles AVL

- Nos interesan funciones para las operaciones de:
  - Pertenencia
  - Inserción
  - Borrado
- Debemos tener en cuenta que tendremos que diseñar funciones auxiliares que permitan realizar estas operaciones manteniendo el árbol equilibrado

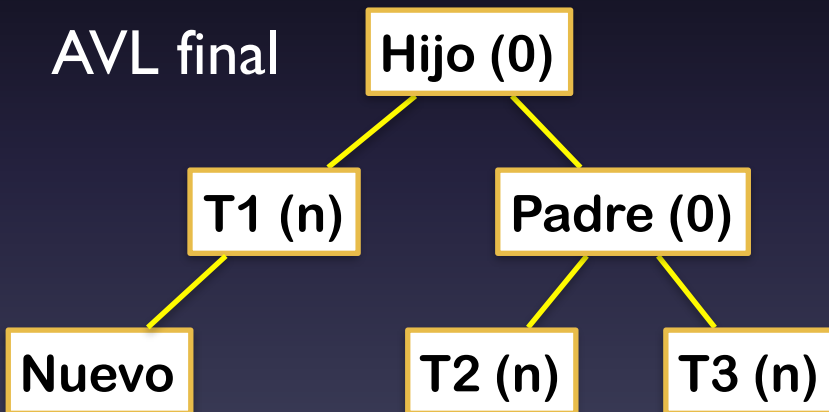
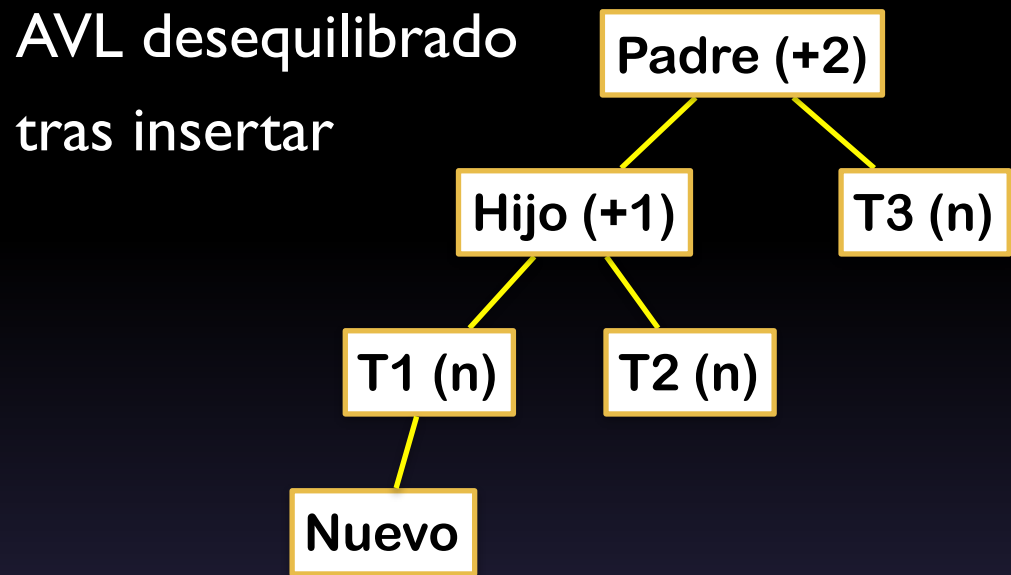
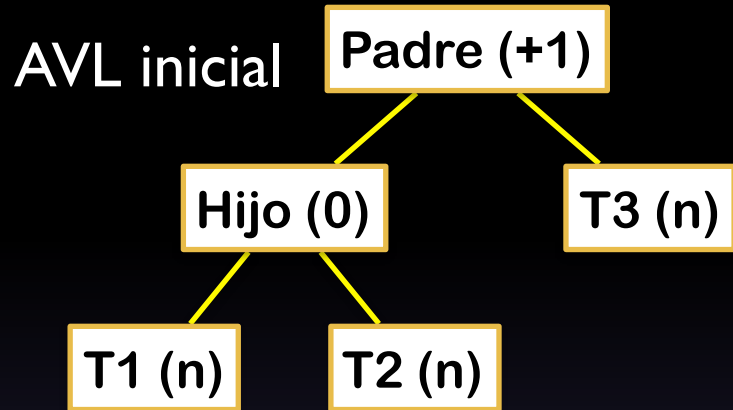
# Equilibrio en inserciones y borrados

- **Idea:** Usar un campo altura en el registro que represente cada uno de los nodos del AVL para determinar el factor de equilibrio (diferencia de altura entre los subárboles izquierdo y derecho), de forma que cuando esa diferencia sea  $>1$  se hagan los reajustes necesarios en los punteros para que tenga una diferencia de alturas  $\leq 1$
- Vamos a verlo en una serie de ejemplos en los que mostraremos todos los casos posibles

# Equilibrio en inserciones y borrados

- Notaremos los subárboles como  $T_k$ , anotando entre paréntesis su altura
- Notaremos el factor de equilibrio como un valor con signo ubicado entre paréntesis junto a cada padre o hijo
- Las dos situaciones posibles que pueden representarse son:
  - Rotaciones simples
  - Rotaciones dobles

# Rotación simple a la derecha

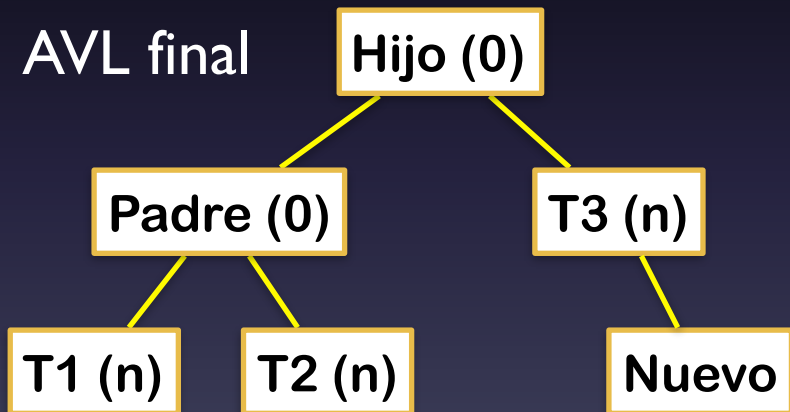
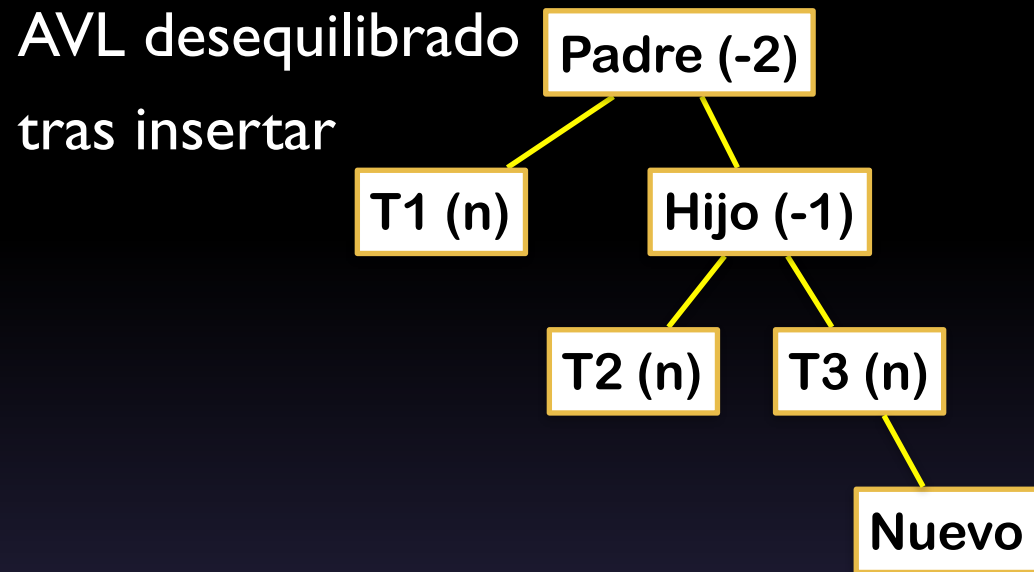
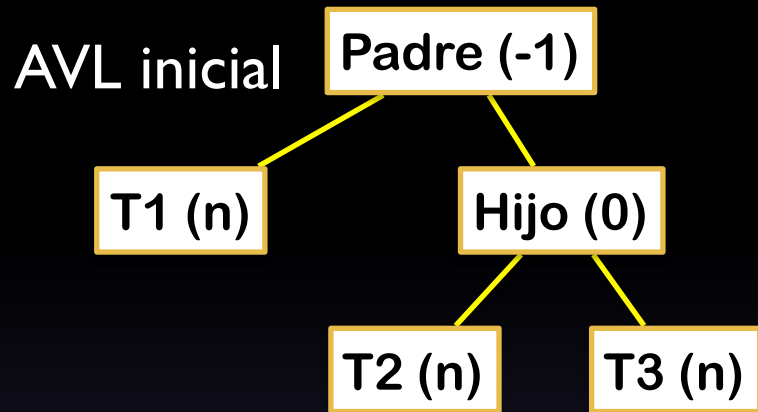


Reajustes necesarios para la rotación  
padre  $\rightarrow$  izqda = hijo  $\rightarrow$  drcha  
hijo  $\rightarrow$  drcha = padre

a) Se preserva el inorden

b) Altura del árbol final = altura árbol inicial

# Rotación simple a la izquierda



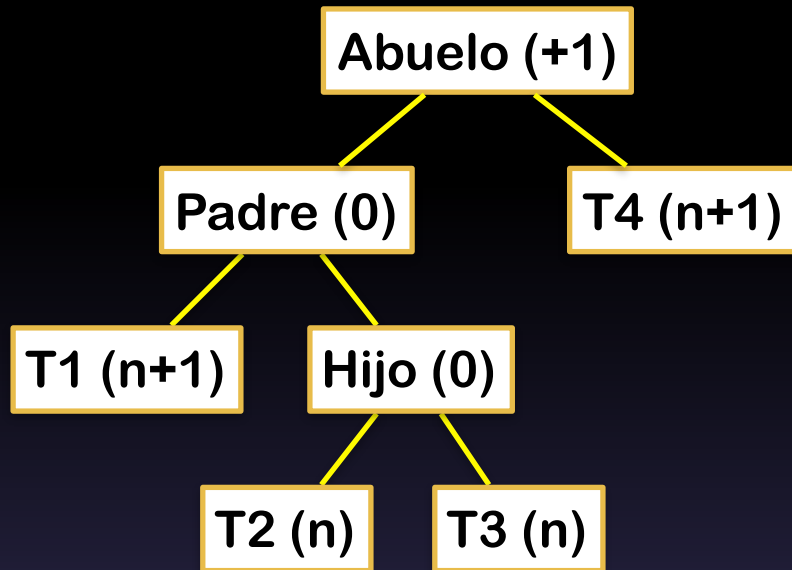
Reajustes necesarios para la rotación  
padre  $\rightarrow$  drcha = hijo  $\rightarrow$  izqda  
hijo  $\rightarrow$  izqda = padre

a) Se preserva el inorden

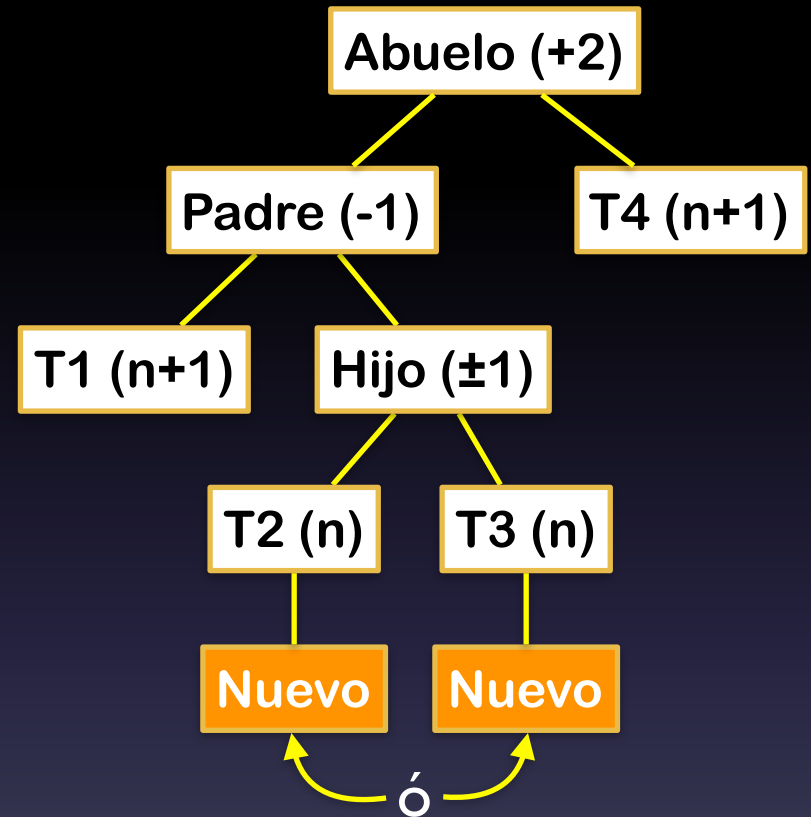
b) Altura del árbol final = altura árbol inicial



# Rotación doble a la derecha

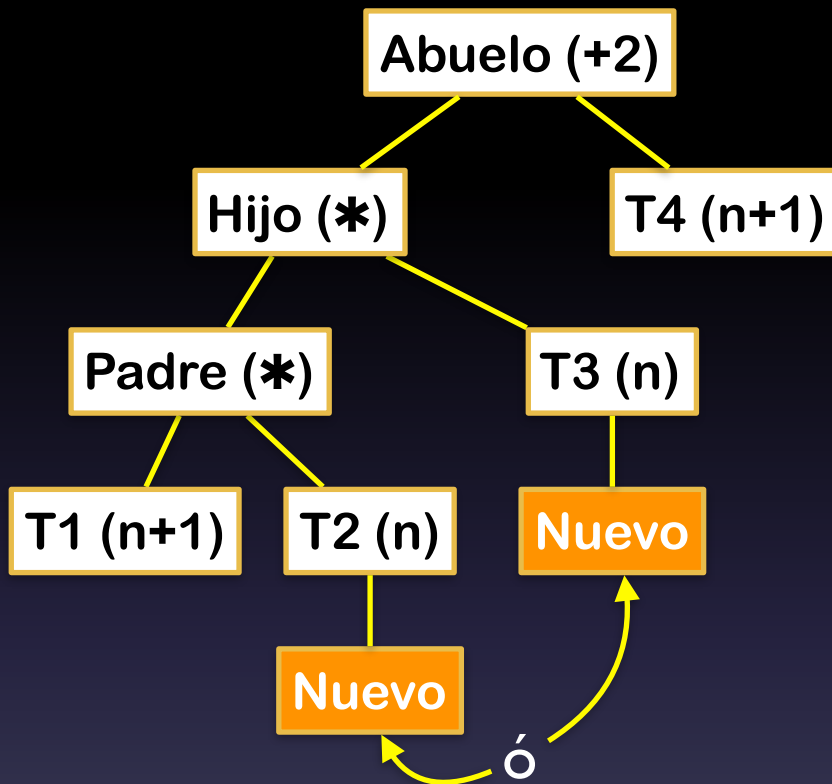


AVL inicial

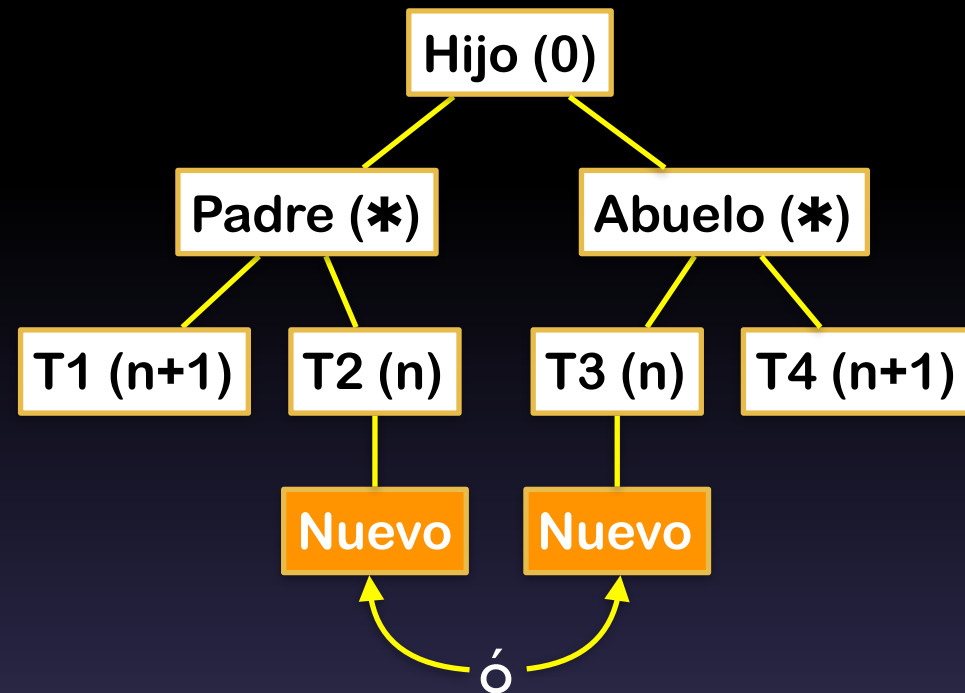


AVL desequilibrado  
tras insertar

# Rotación doble a la derecha

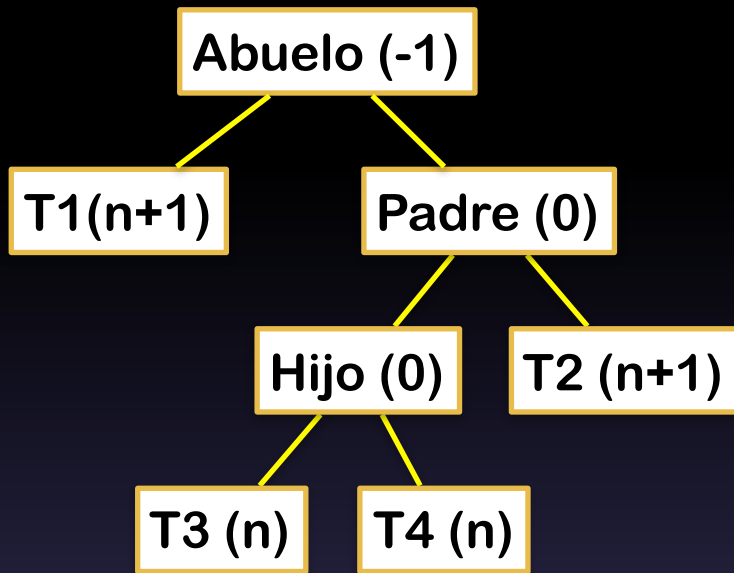


Rotación simple a izquierda  
en padre

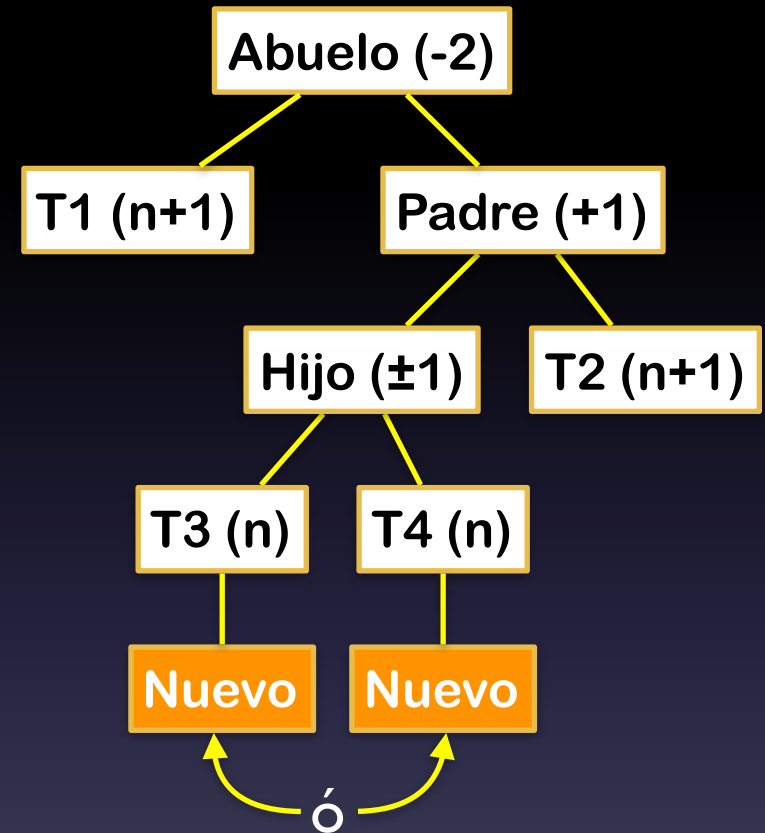


Rotación simple a derecha  
en abuelo

# Rotación doble a la izquierda

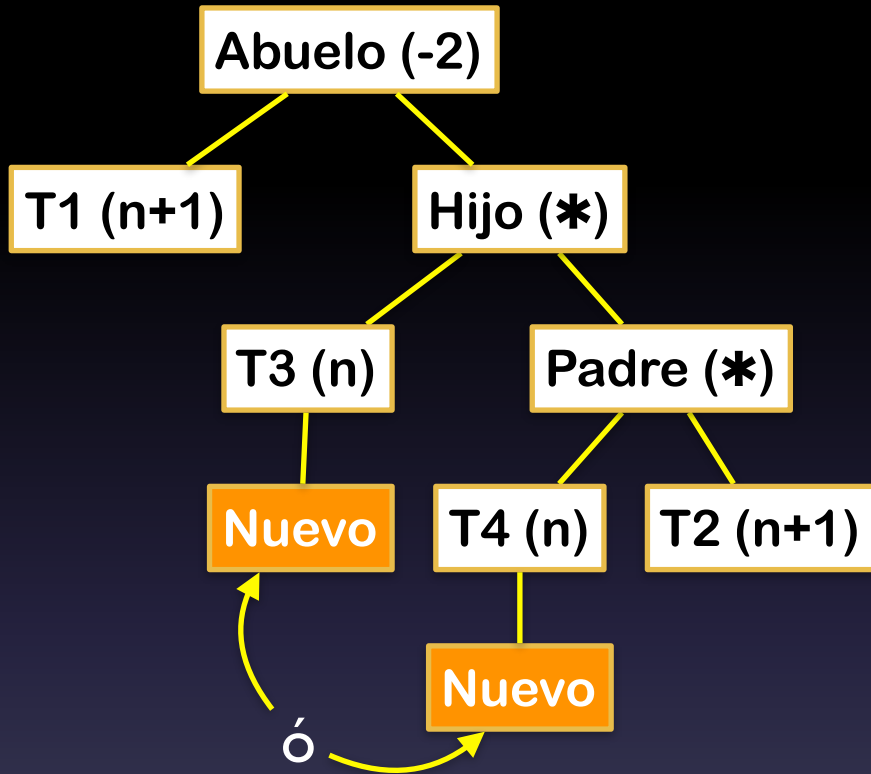


AVL inicial

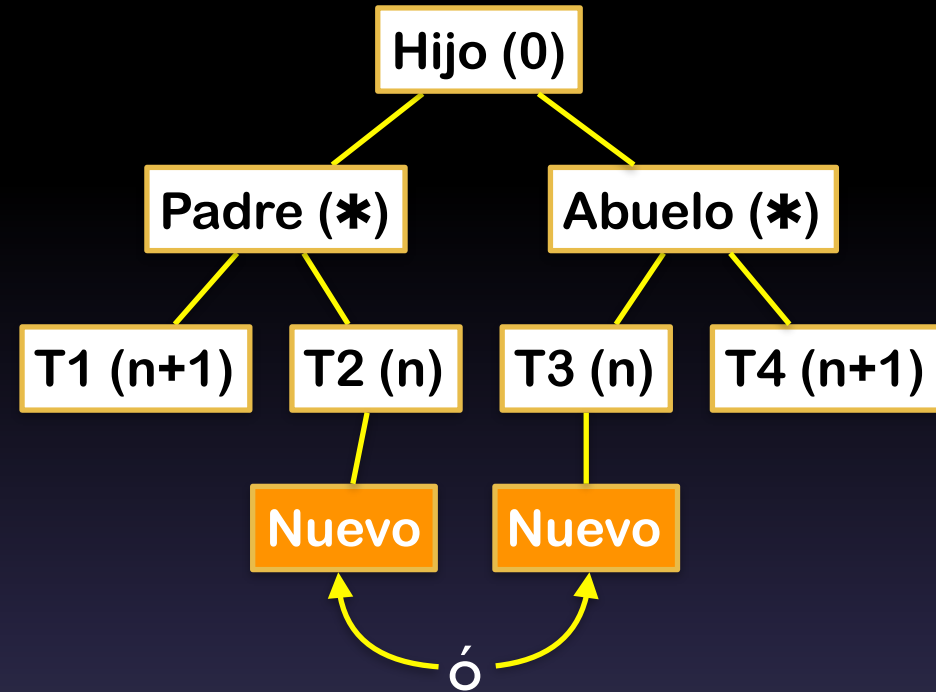


AVL desequilibrado  
tras insertar

# Rotación doble a la izquierda



Rotación simple a derecha  
en padre



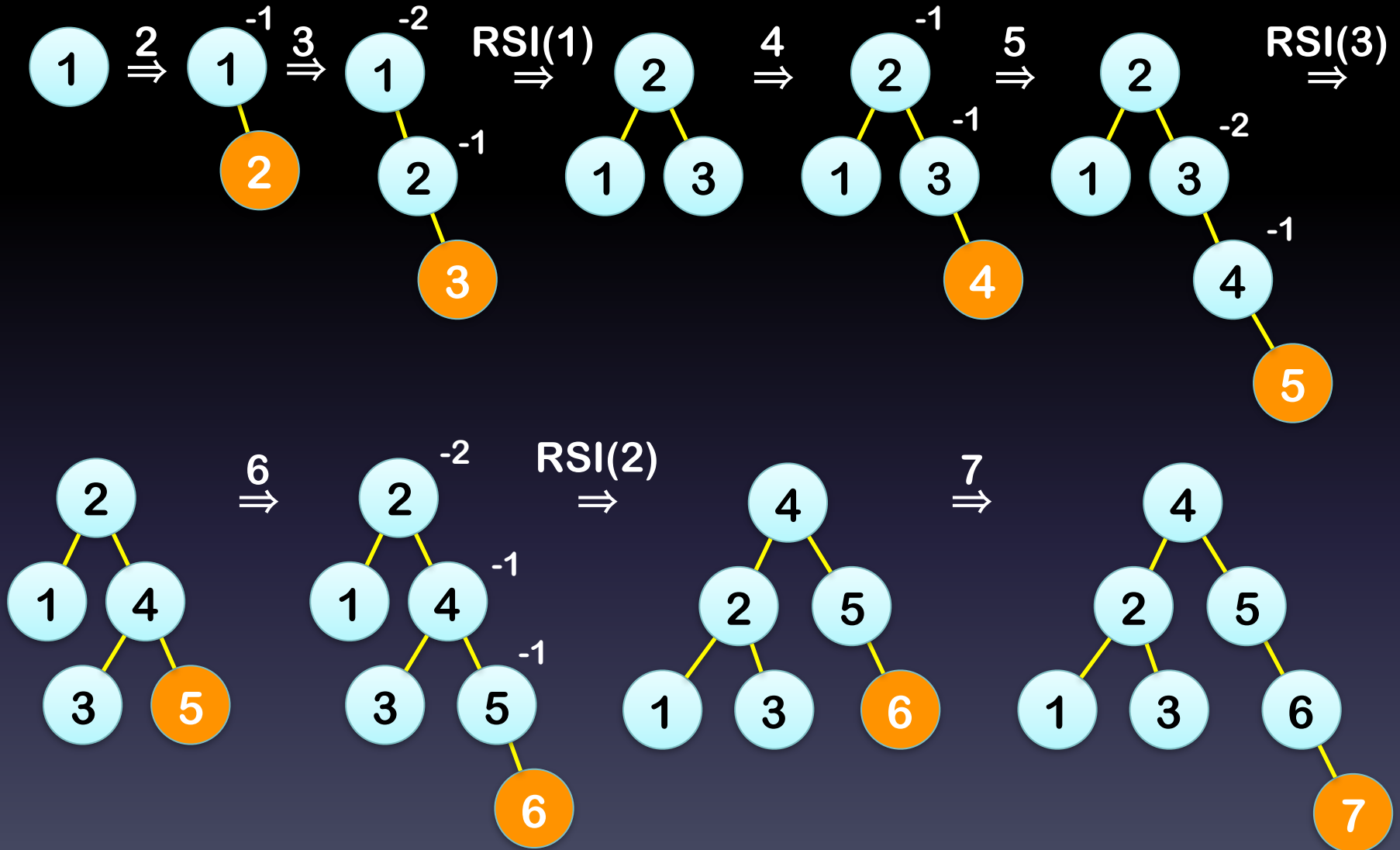
Rotación simple a izquierda  
en abuelo

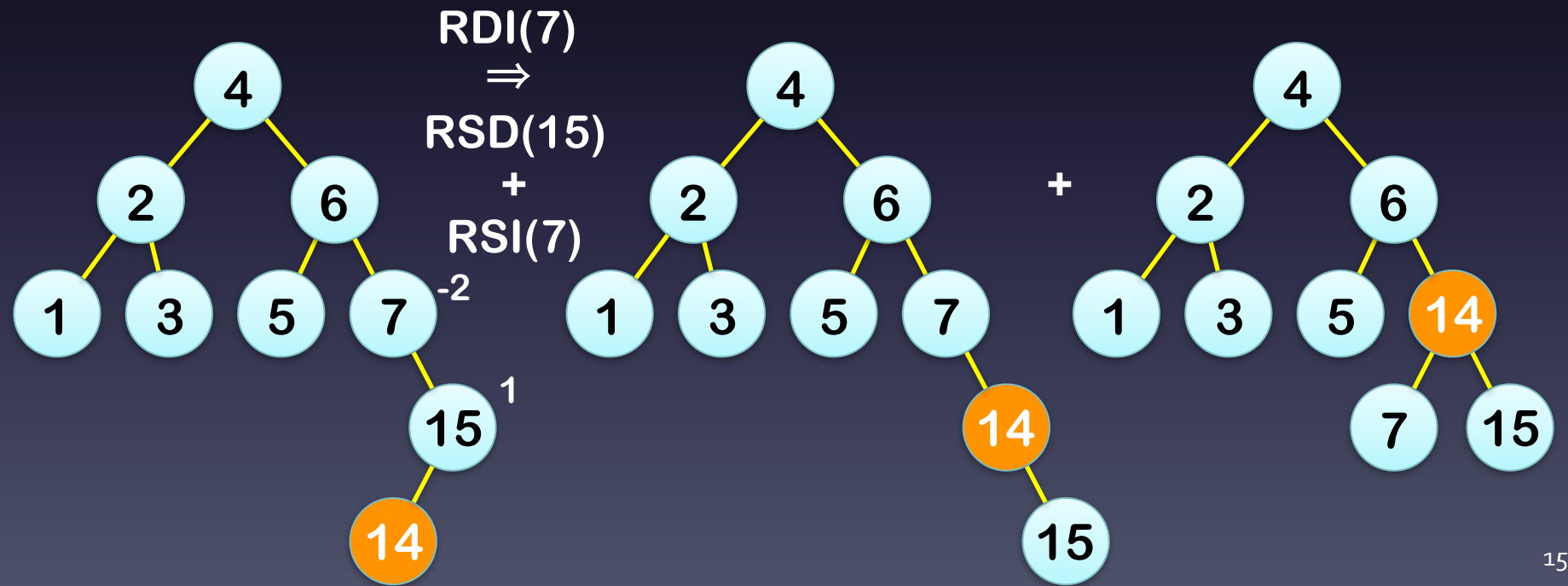
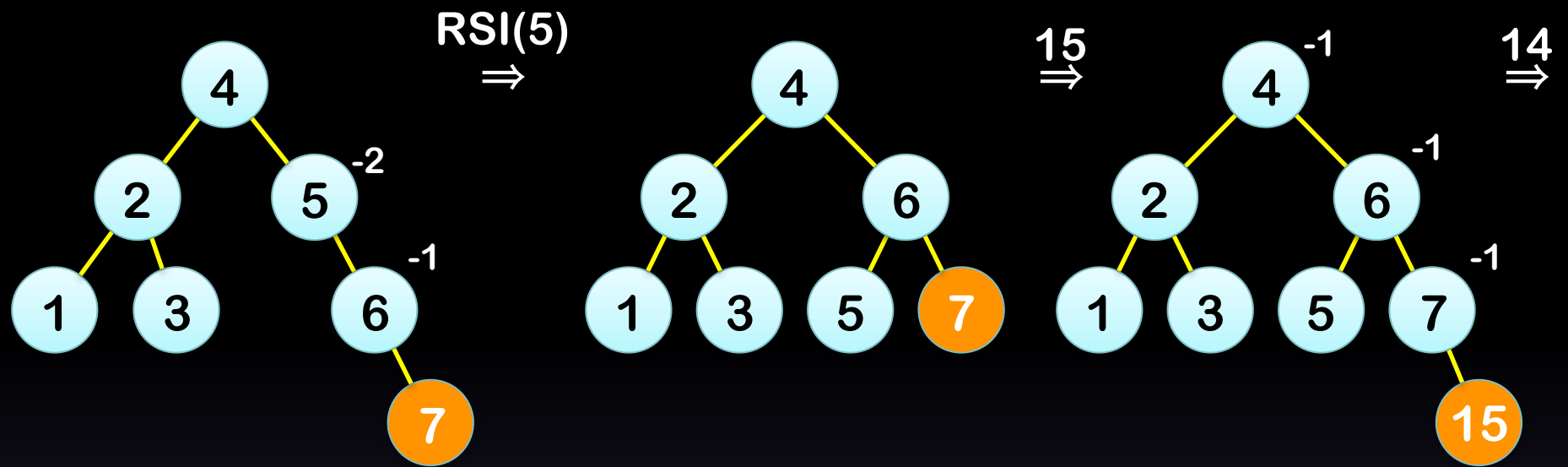
# ¿Qué rotación utilizar?

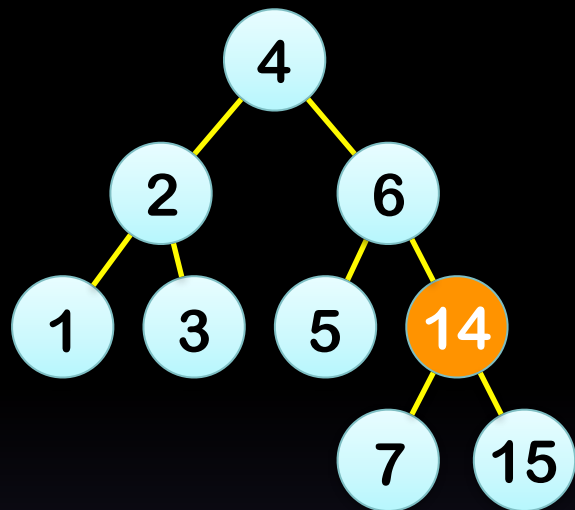
Si la inserción se realiza en:

- el hijo izquierdo del hijo izquierdo del nodo desequilibrado  $\Rightarrow$  RSD
- el hijo derecho del hijo derecho del nodo desequilibrado  $\Rightarrow$  RSI
- el hijo derecho del hijo izquierdo del nodo desequilibrado  $\Rightarrow$  RDD
- el hijo izquierdo del hijo derecho del nodo desequilibrado  $\Rightarrow$  RDI

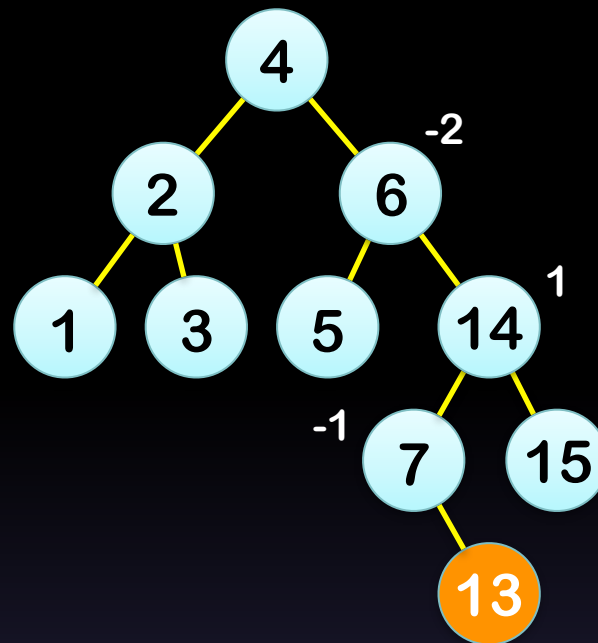
# Ejemplo



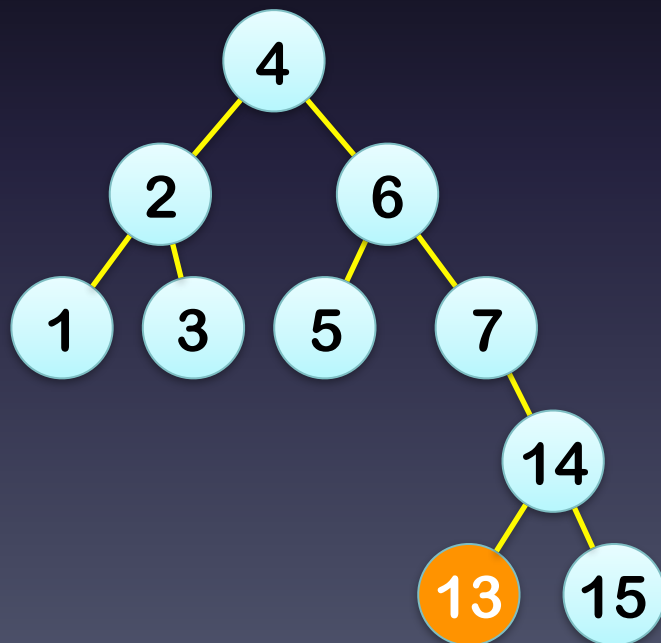




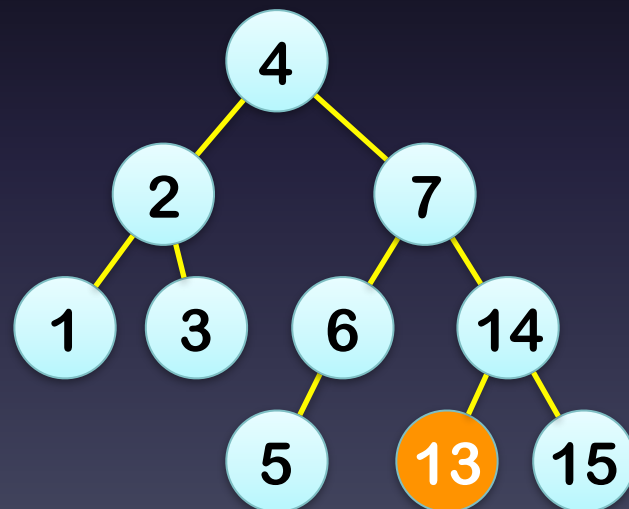
13  
⇒



RDI(6)  
⇒  
RSD(14)  
+  
RSI(6)



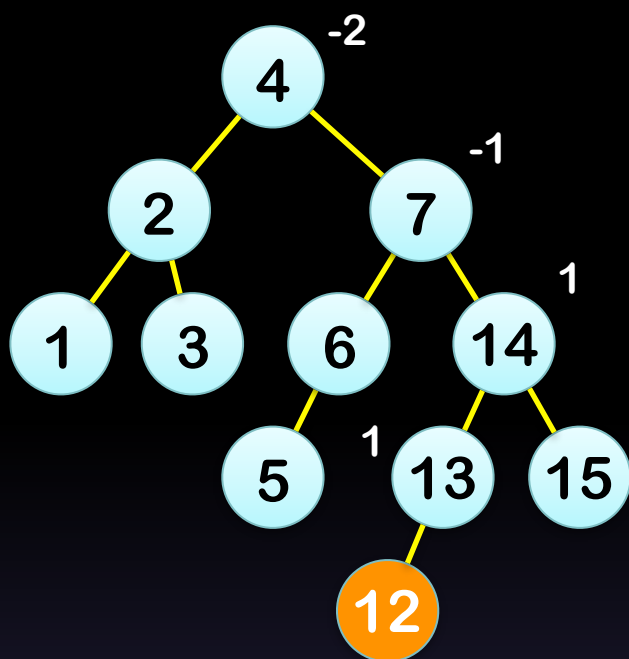
+



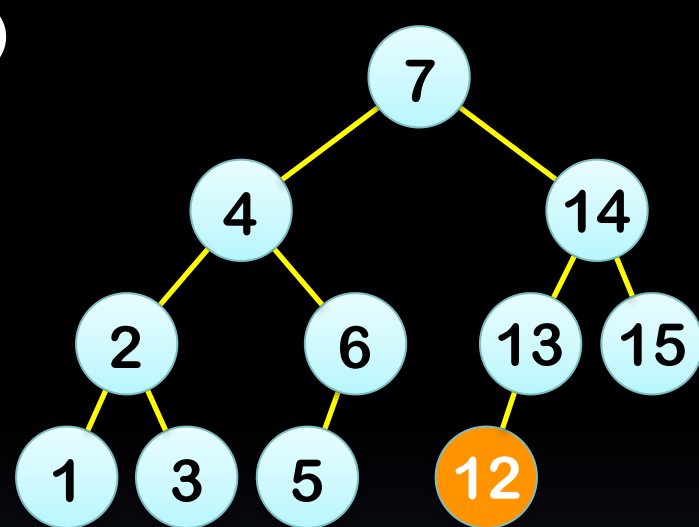
12  
⇒



12  
⇒

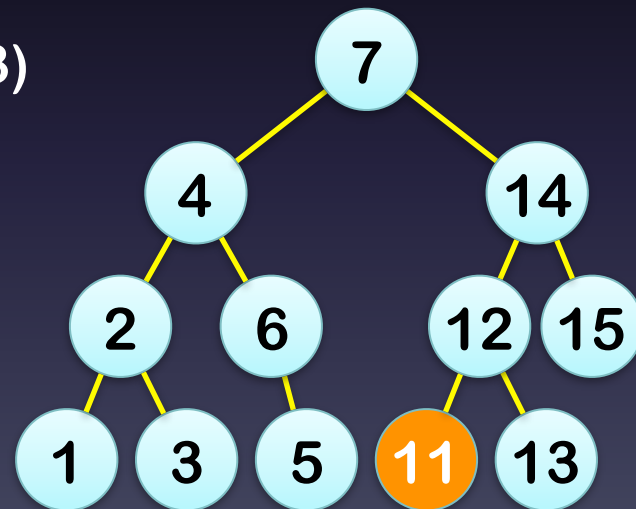
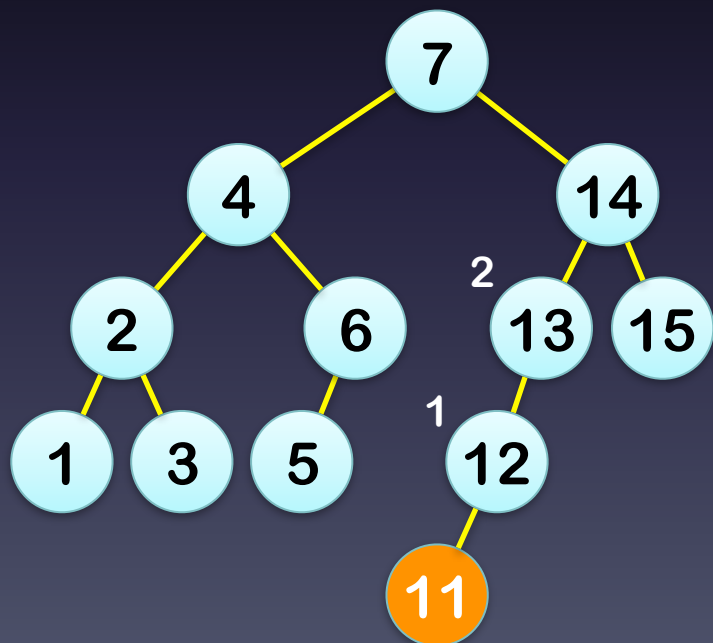


RSI(4)  
⇒



11  
⇒

RSD(13)  
⇒

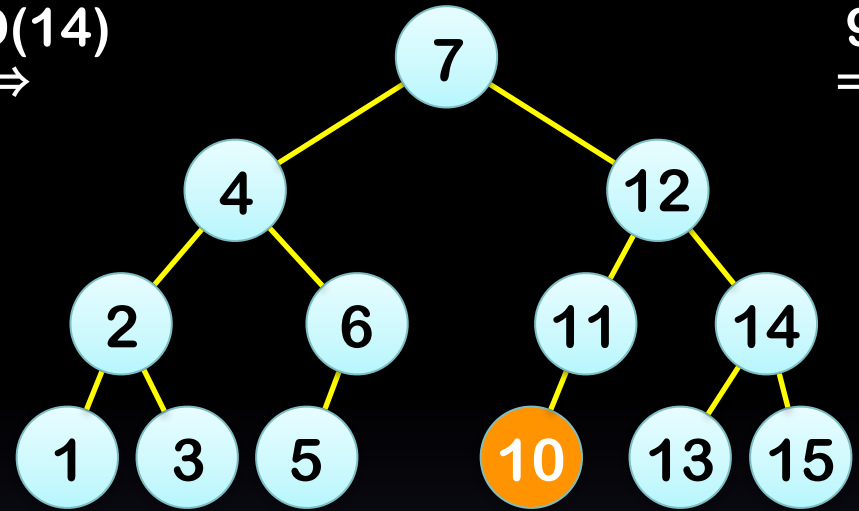
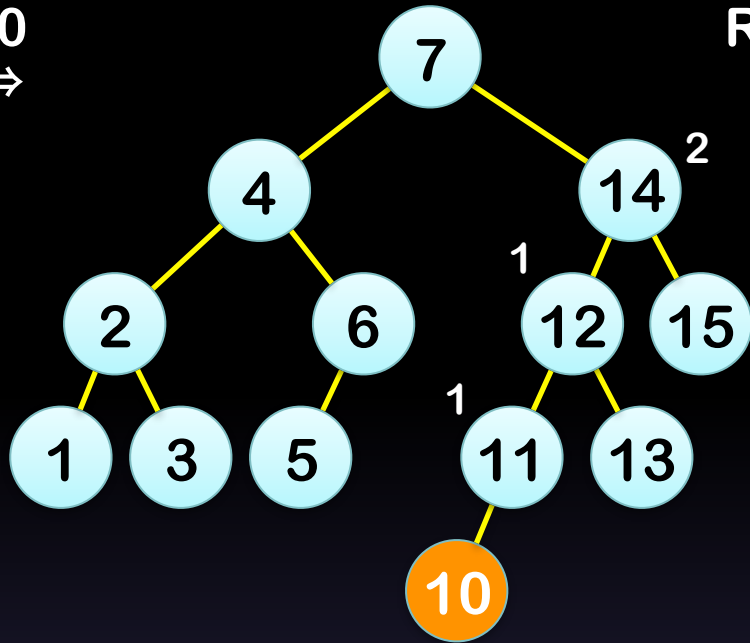


10  
⇒

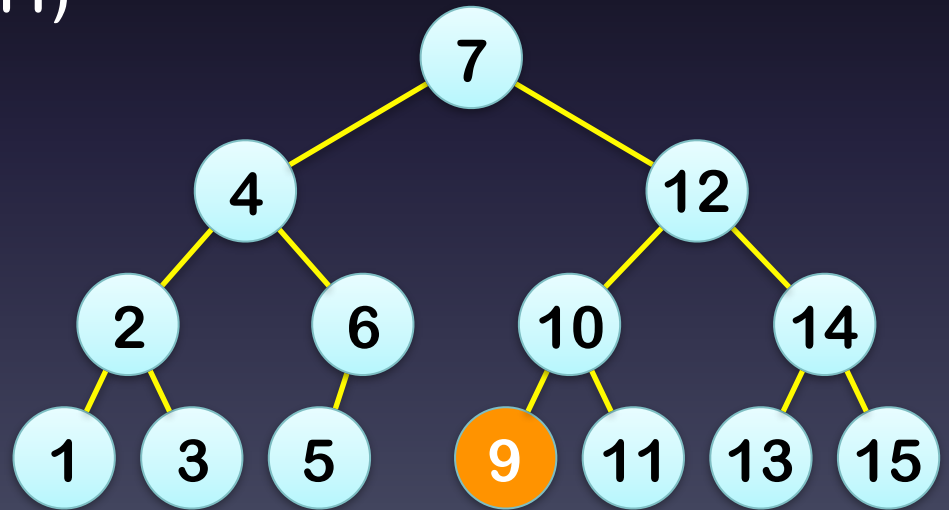
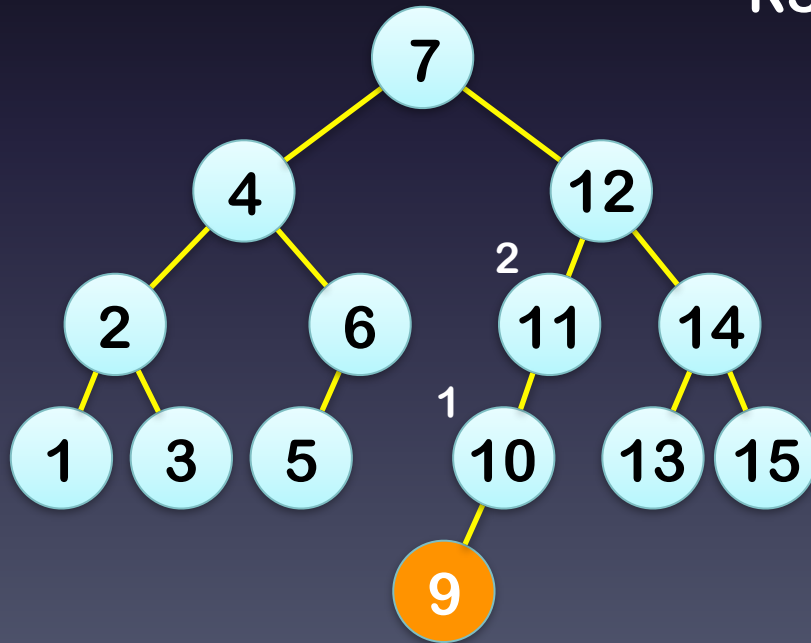
10  
⇒

RSD(14)  
⇒

9  
⇒



RSD(11)  
⇒



AVL Final