

Diccionarios

Un diccionario es una estructura de datos que permite las siguientes operaciones:

- Asociar un **valor** a una **clave**, o actualizarlo
- Obtener el **valor** asociado a una **clave**

Y para ciertos casos de uso:

- Eliminar del diccionario una **clave** y su **valor** asociado

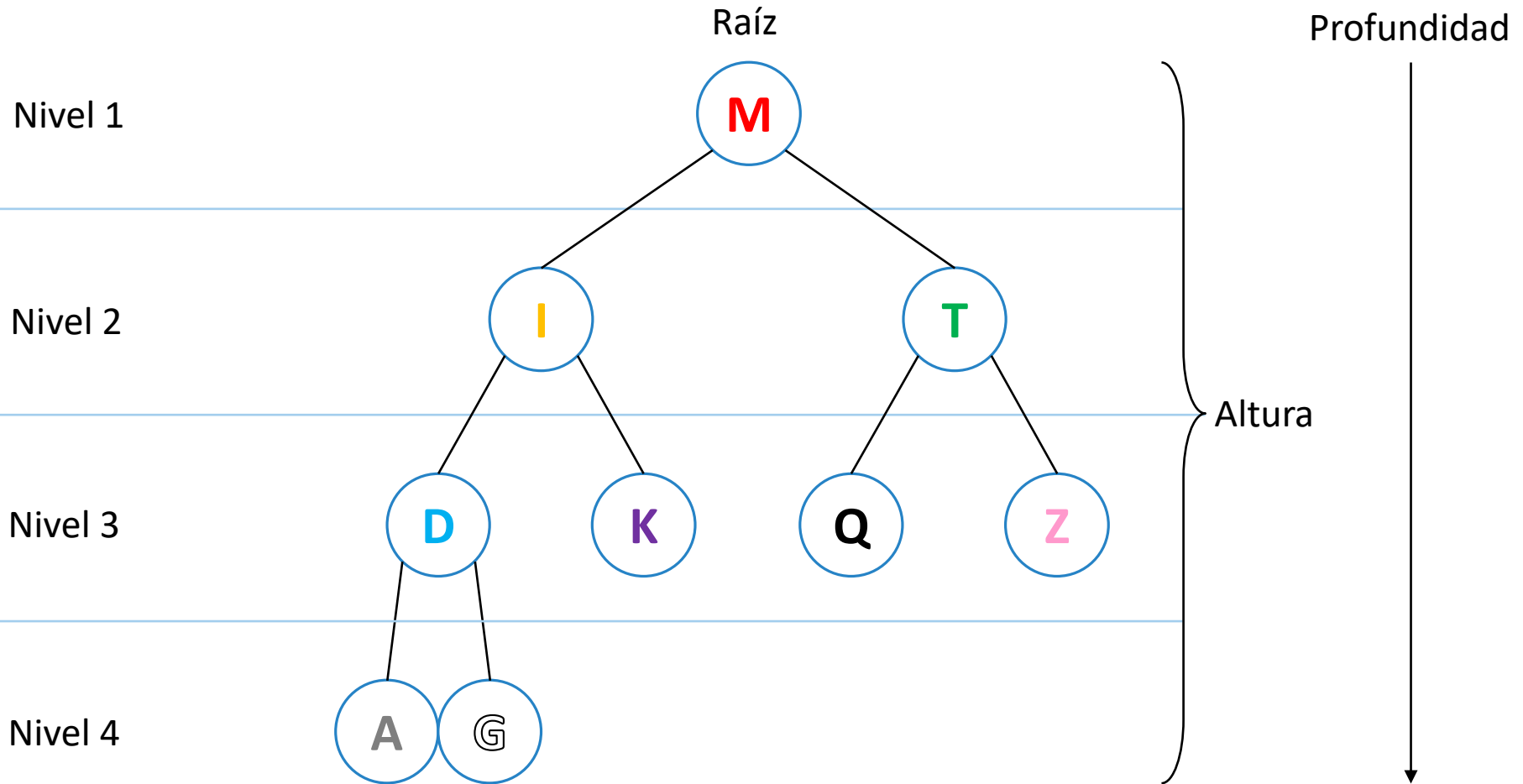
El árbol binario de búsqueda (ABB)

Es un diccionario que guarda tuplas (*key, value*) organizadas en nodos de forma recursiva

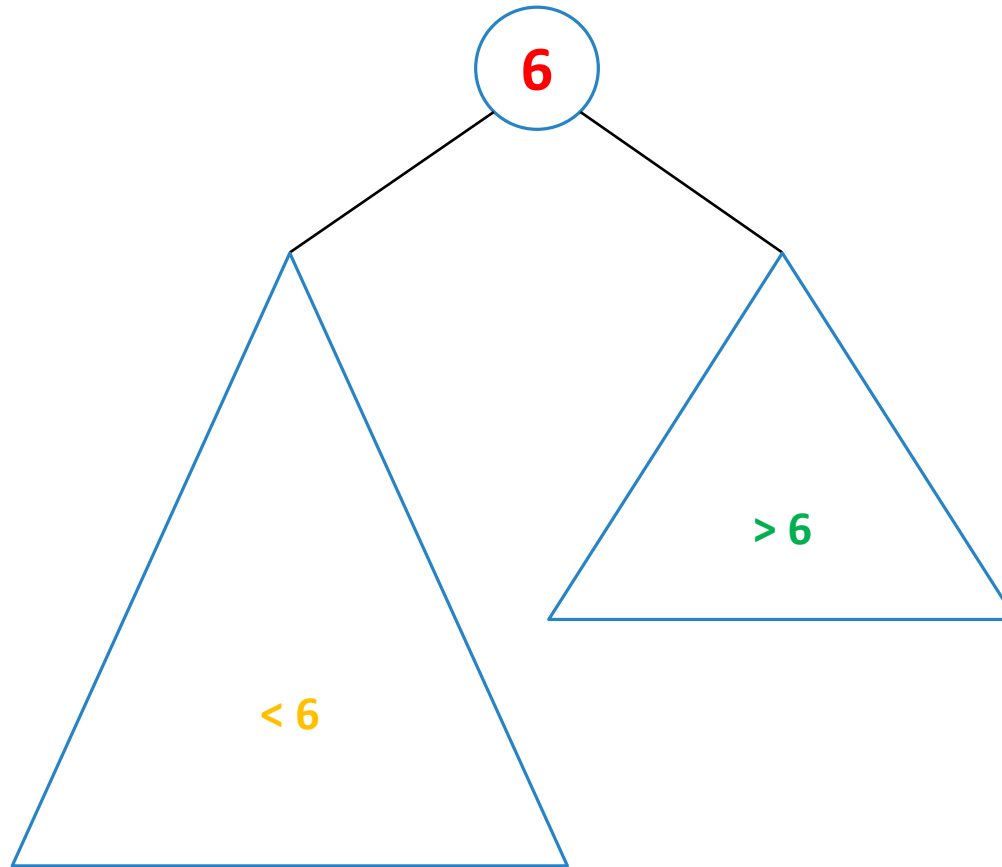
La raíz del árbol almacena una tupla y el resto se organizan recursivamente en un ABB como hijo izquierdo o derecho de la raíz

Los menores cuelgan del hijo izquierdo, y los mayores, del hijo derecho

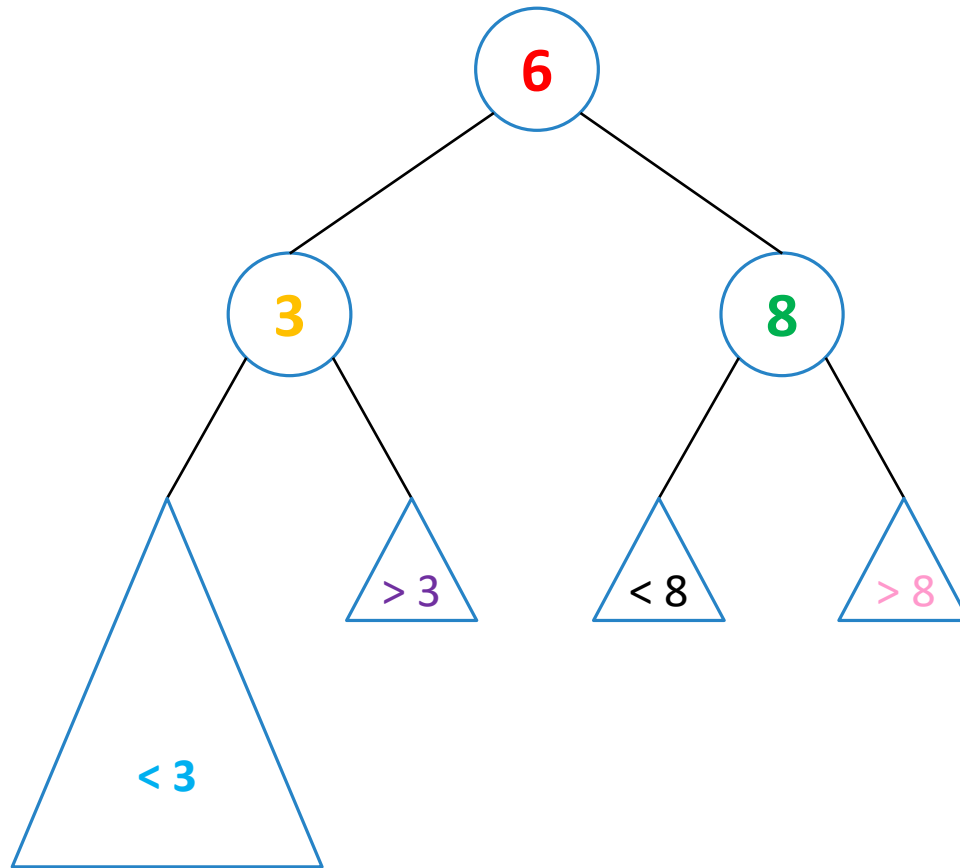
Anatomía de un árbol binario



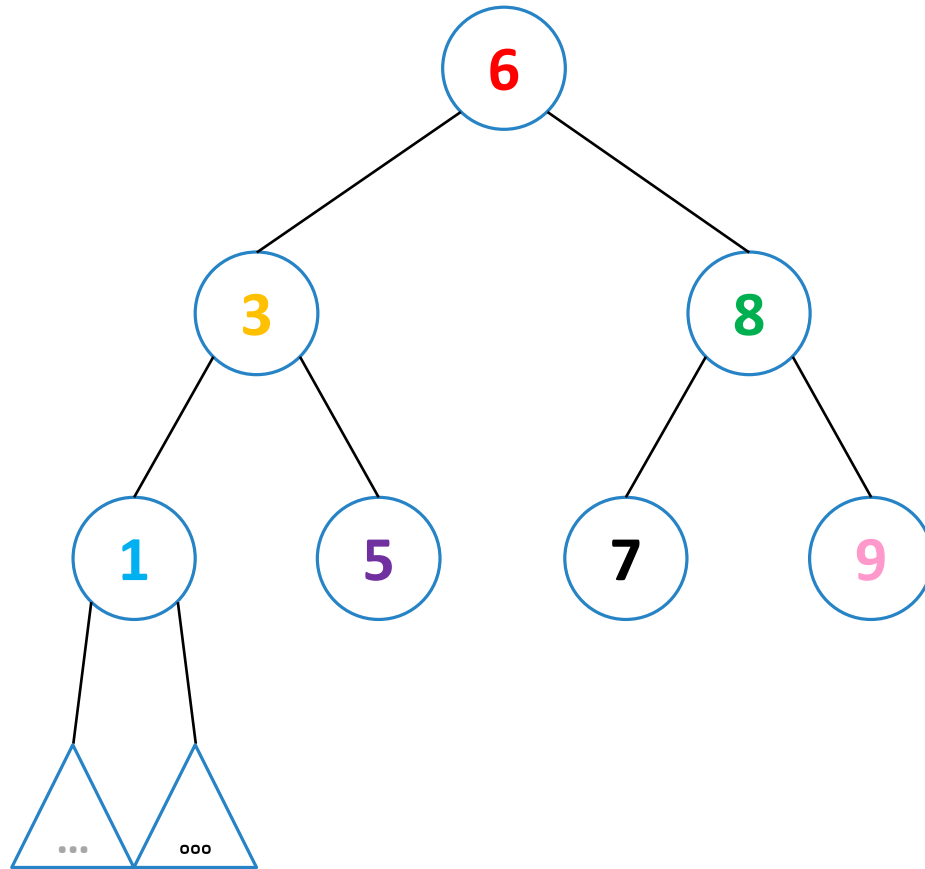
El árbol binario de búsqueda ...



... está compuesto por árboles binarios de búsqueda



... y así hasta las hojas



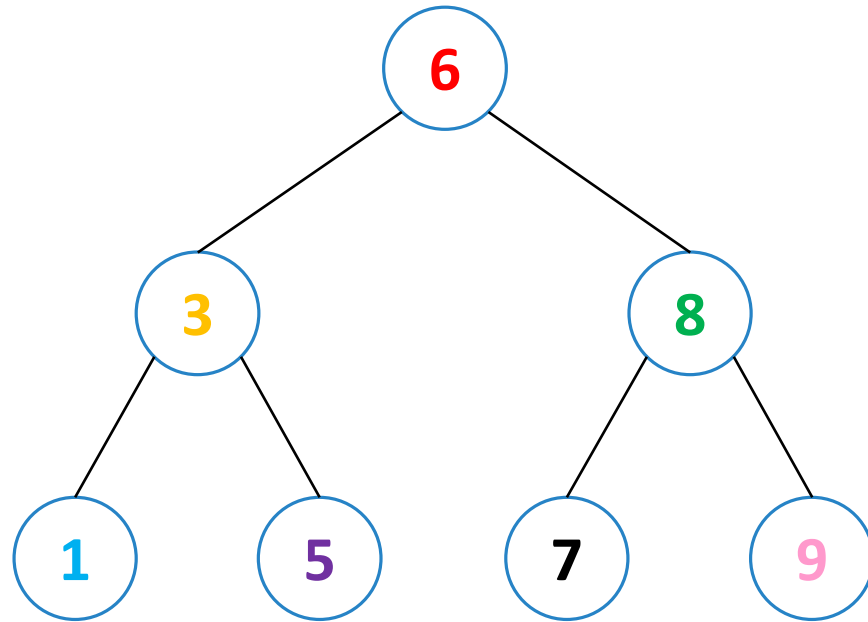
Operaciones del ABB



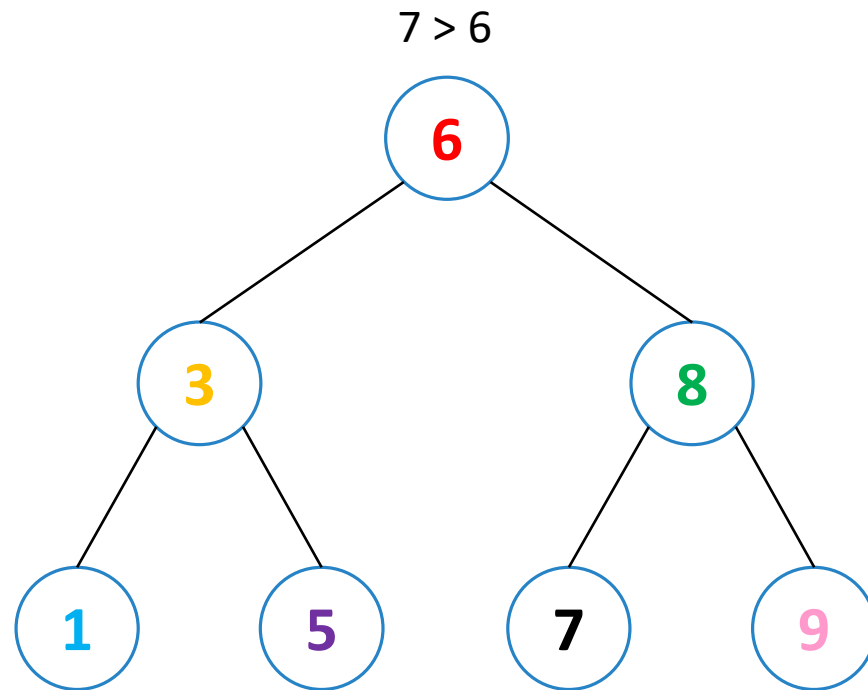
¿Cómo se busca un elemento en el árbol?

Tratemos de aprovechar que la estructura es recursiva

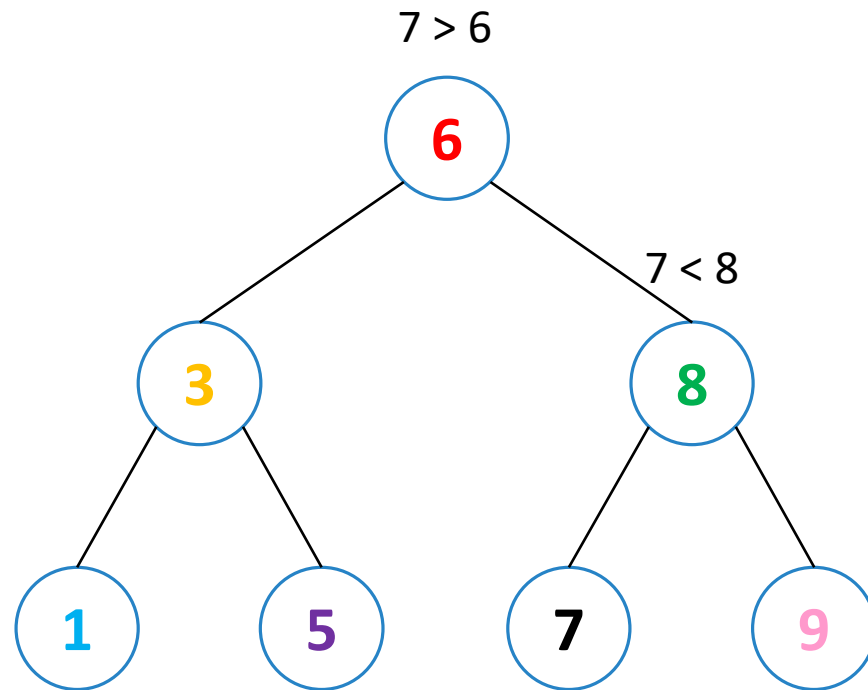
Busquemos el 7



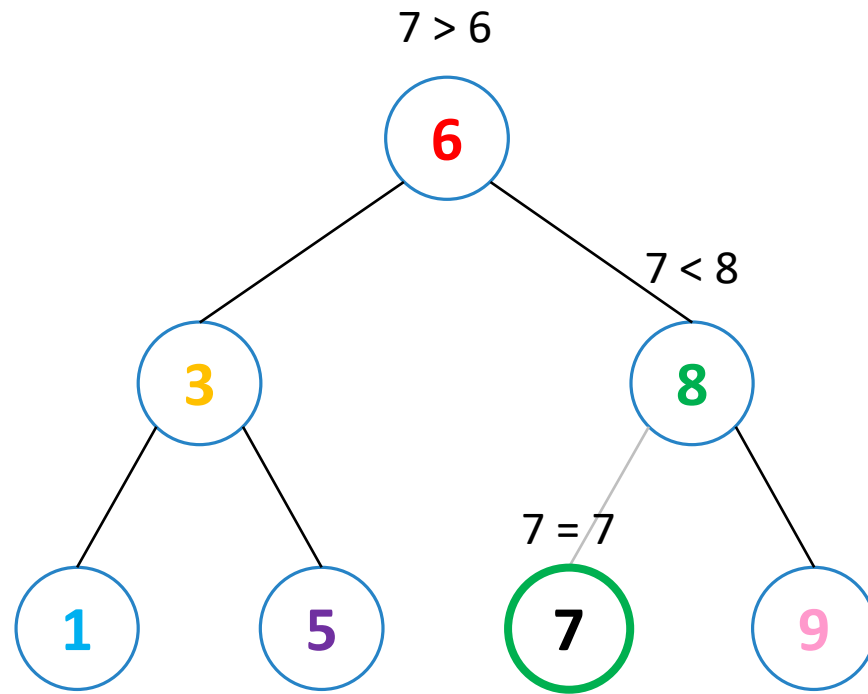
Busquemos el 7



Busquemos el 7



Busquemos el 7



search(A, k):

if $A = \emptyset$ *o* $A.key = k$:

return A

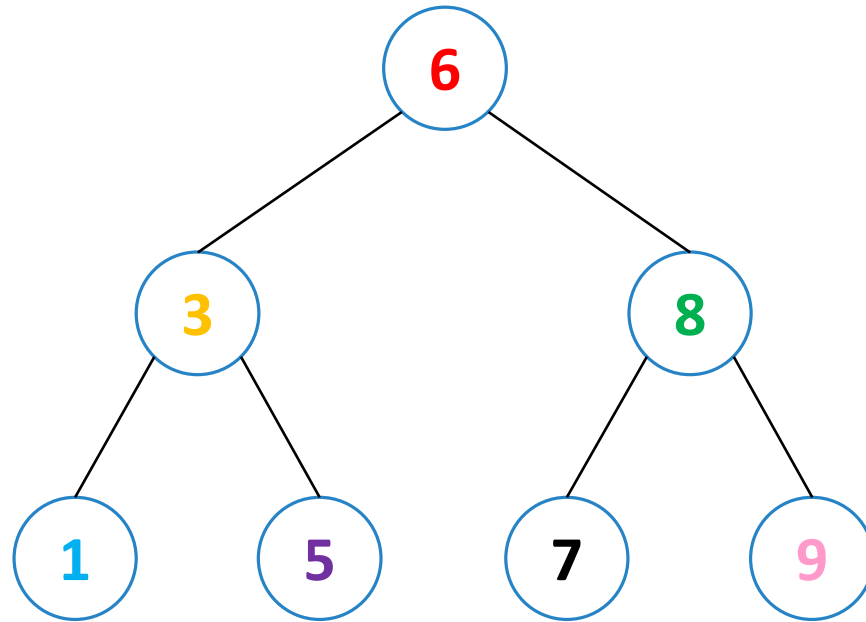
else if $k < A.key$:

search($A.left, k$)

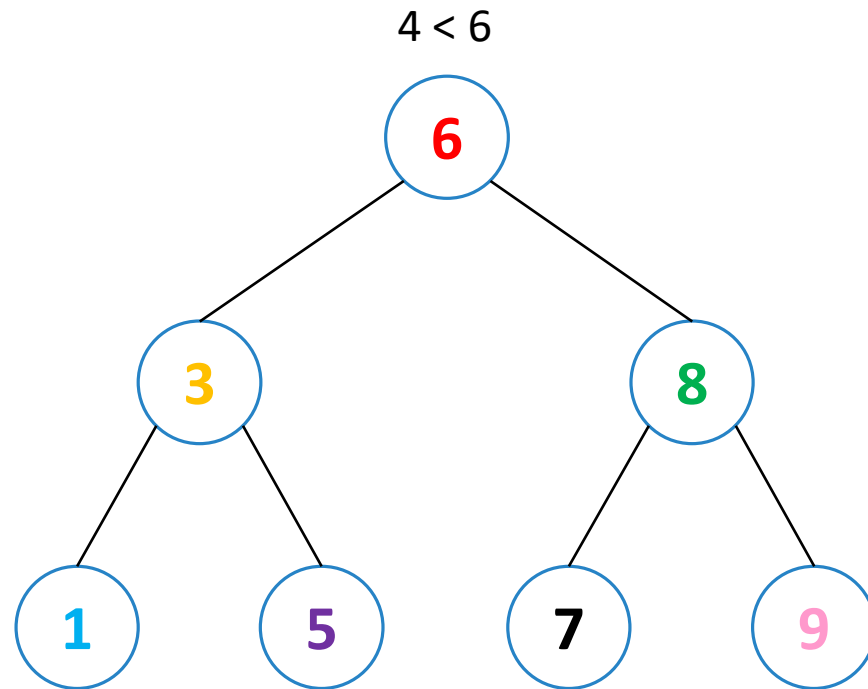
else:

search($A.right, k$)

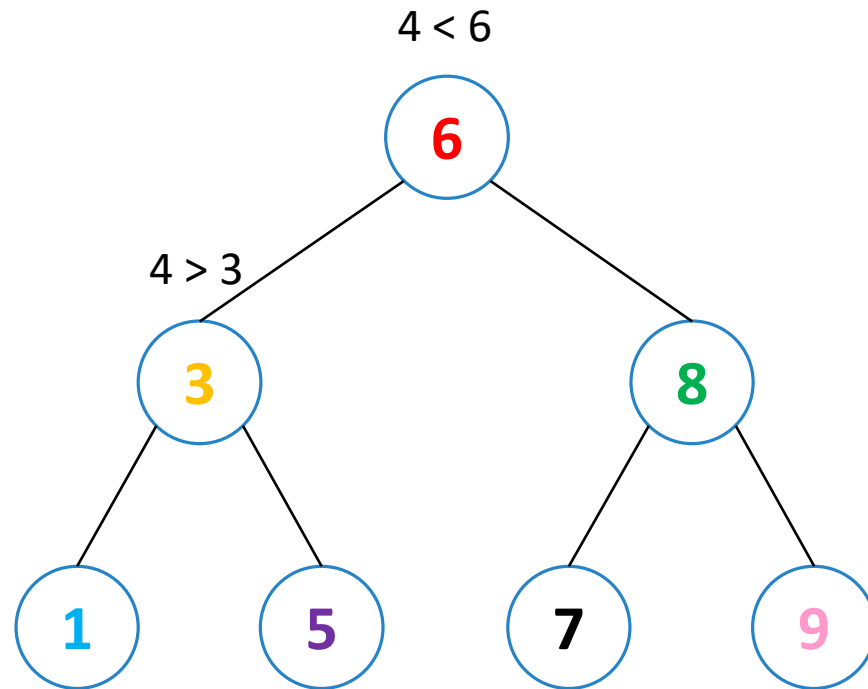
Insertemos el 4



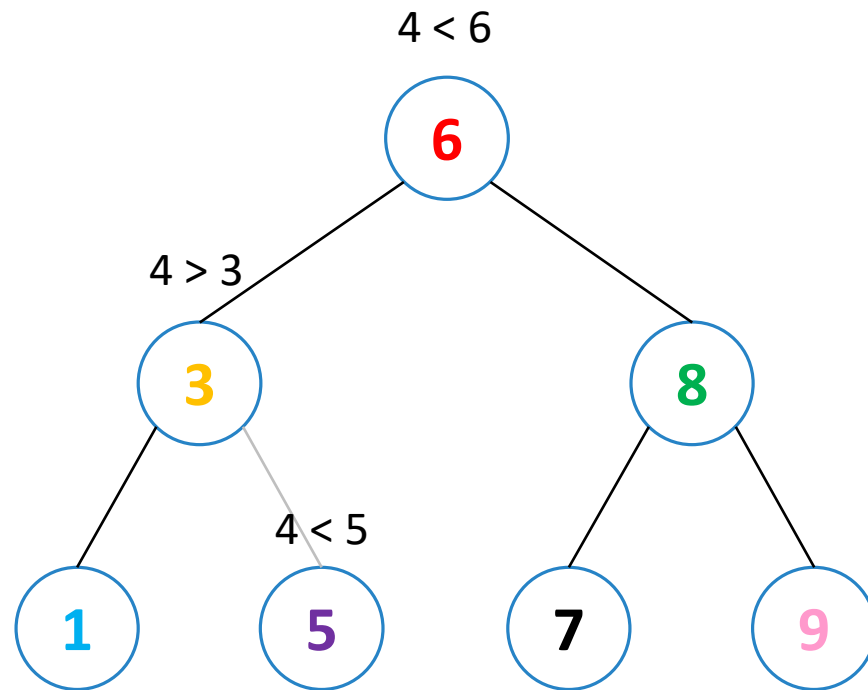
Insertemos el 4



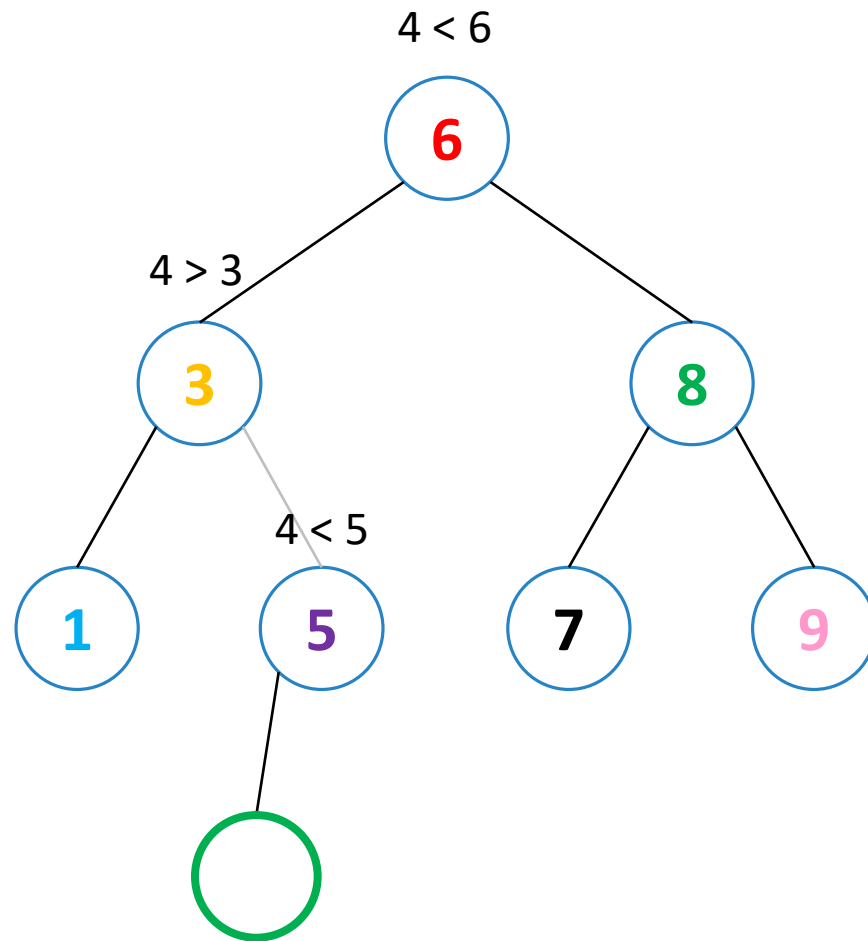
Insertemos el 4



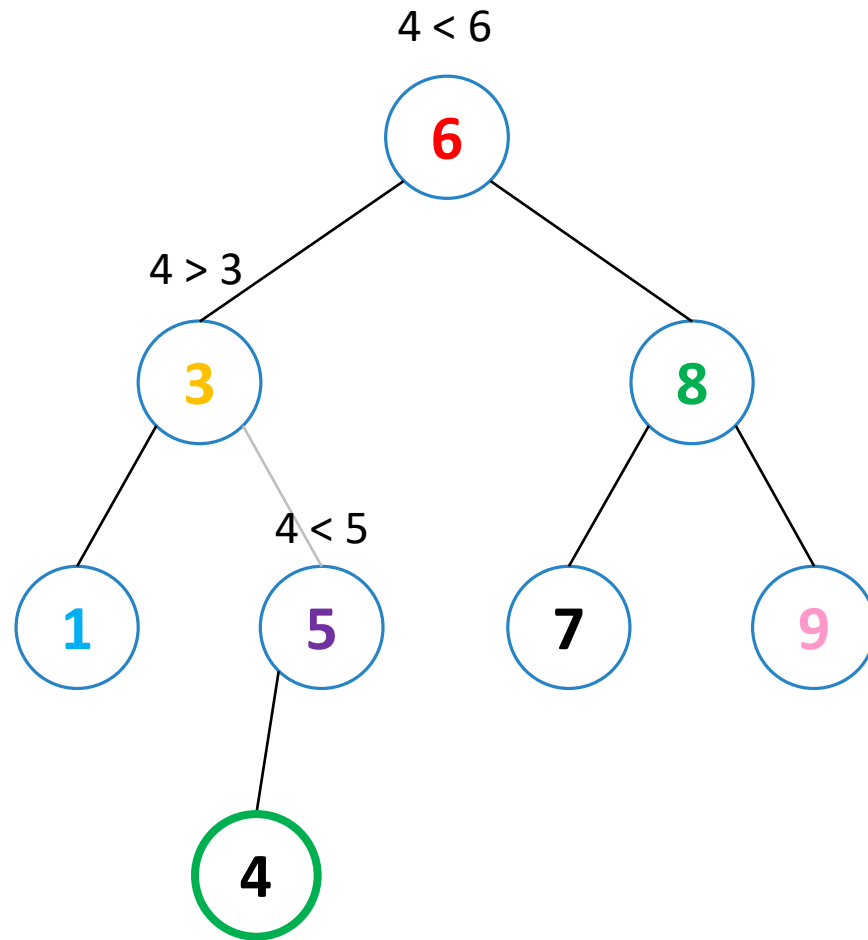
Insertemos el 4



Insertemos el 4



Insertemos el 4



insert(A, k):

$B \leftarrow \textit{search}(A, k)$

$B.\textit{key} \leftarrow k$

Eliminación



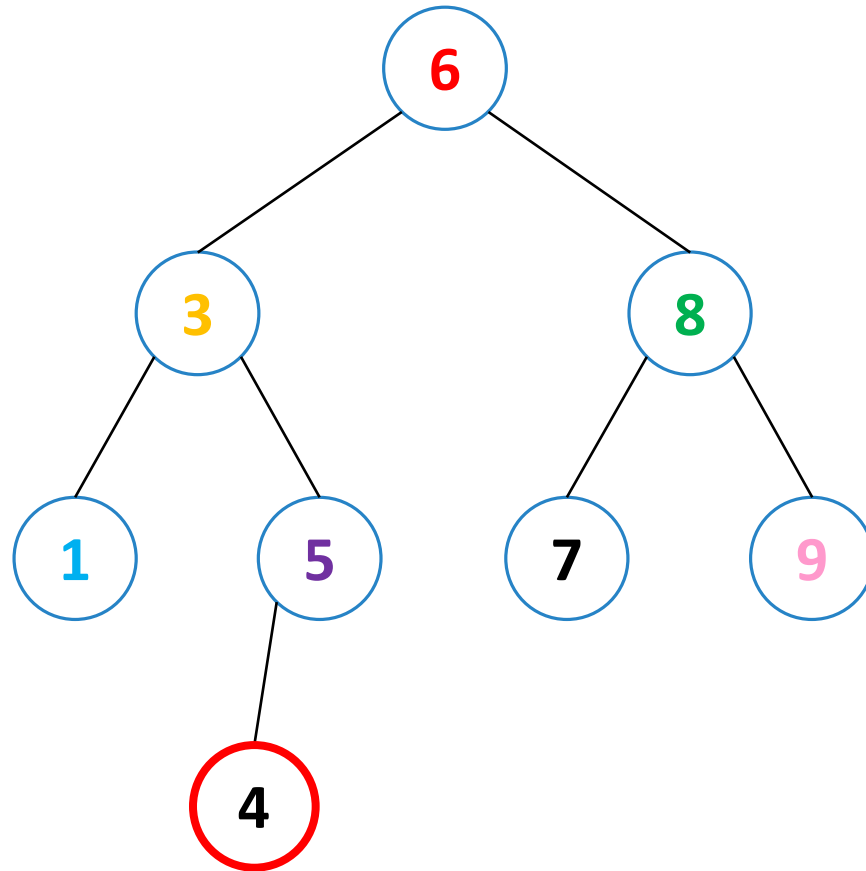
Queremos eliminar un dato del árbol

Si el dato quedó en una hoja, o tiene un solo hijo, eliminarlo es trivial

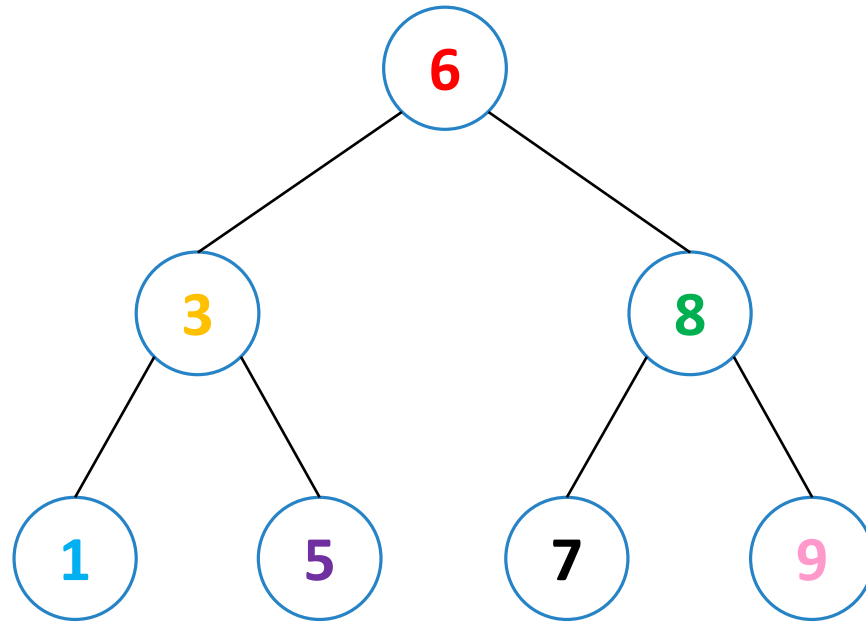
Si no, ¿cómo podemos eliminarlo sin romper la estructura?

¿Podremos reemplazarlo por otro nodo del árbol? ¿Cuál?

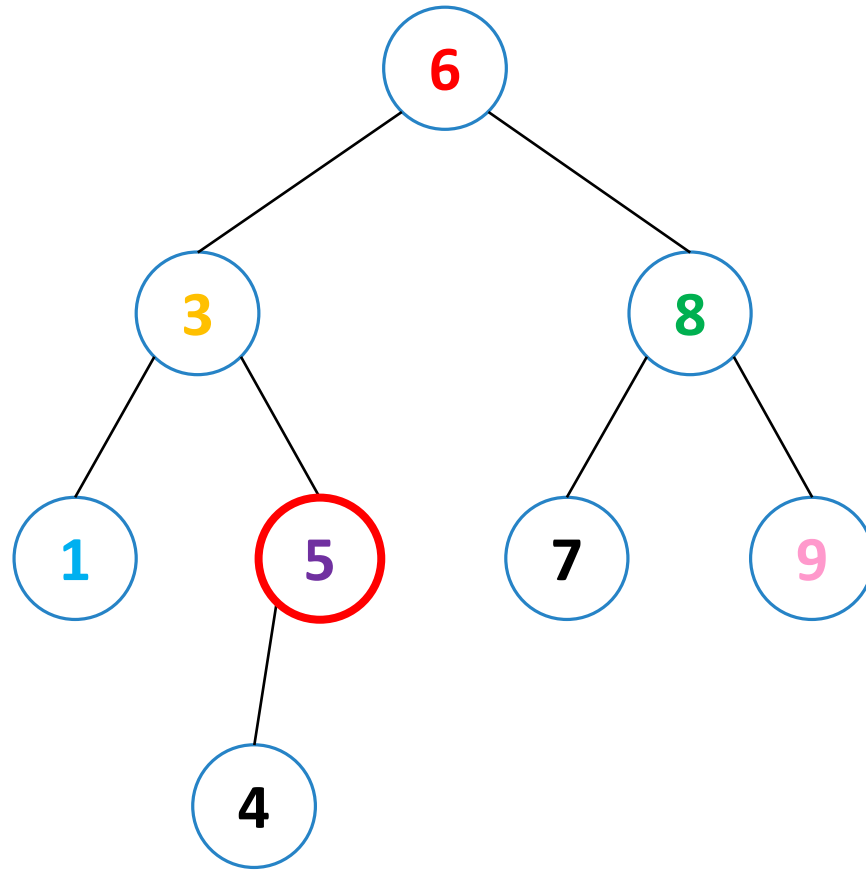
Eliminemos el 4



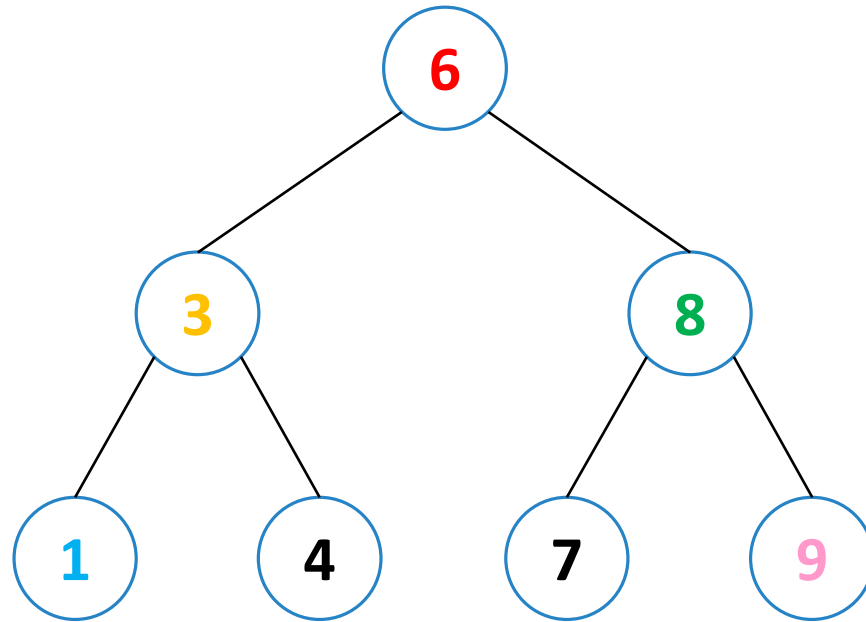
Eliminemos el 4



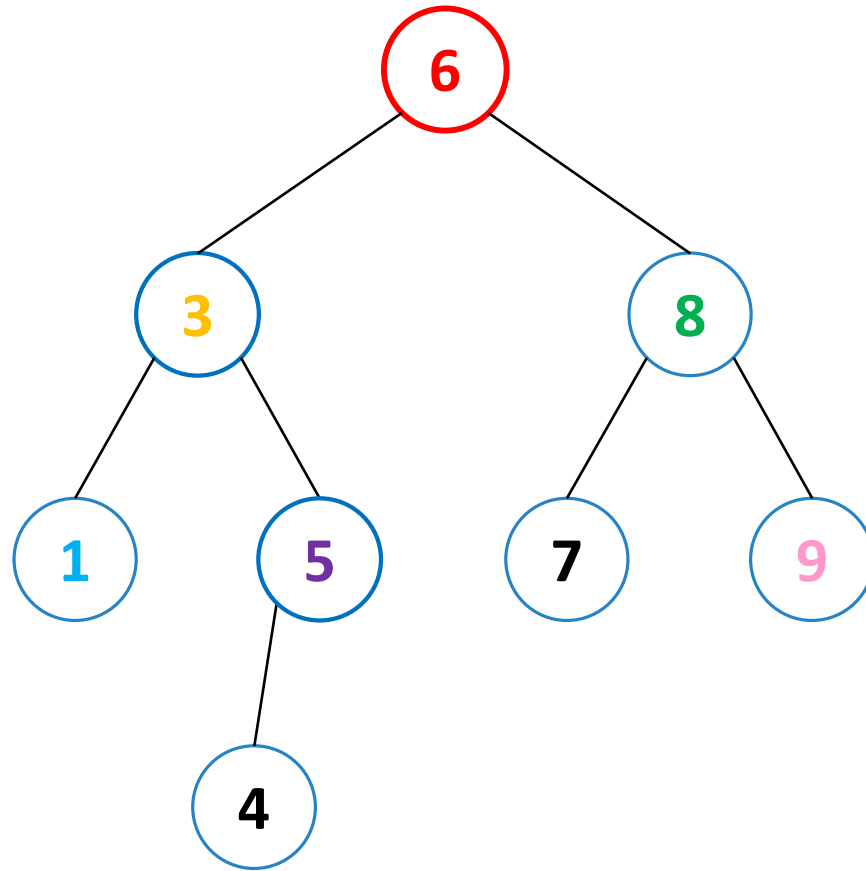
Eliminemos el 5



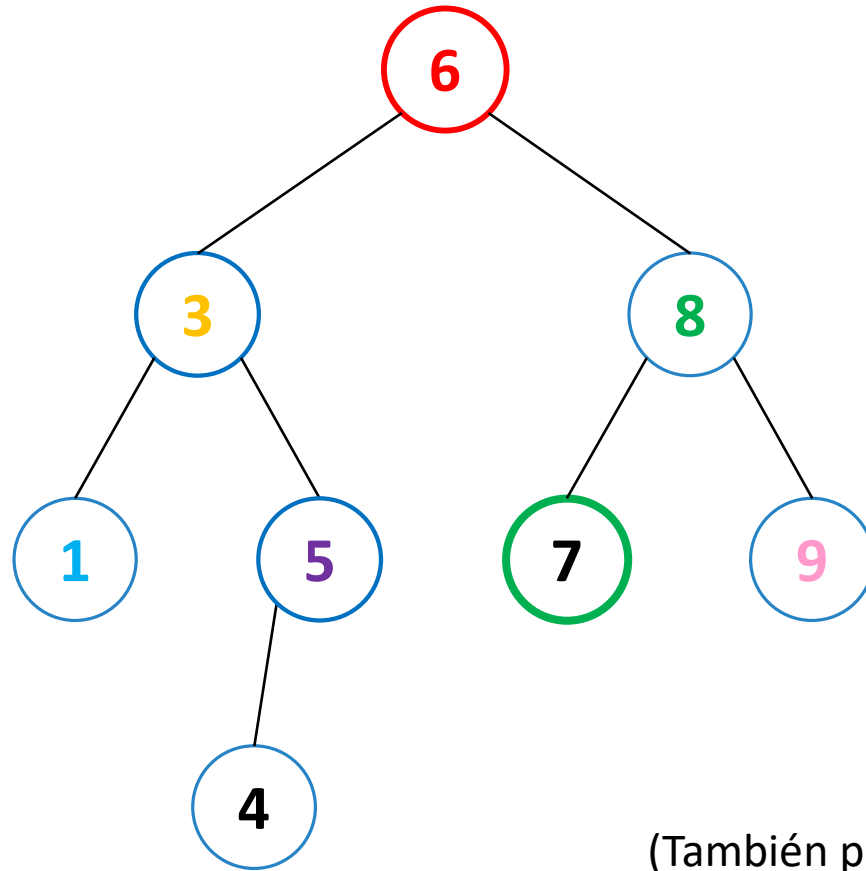
Eliminemos el 5



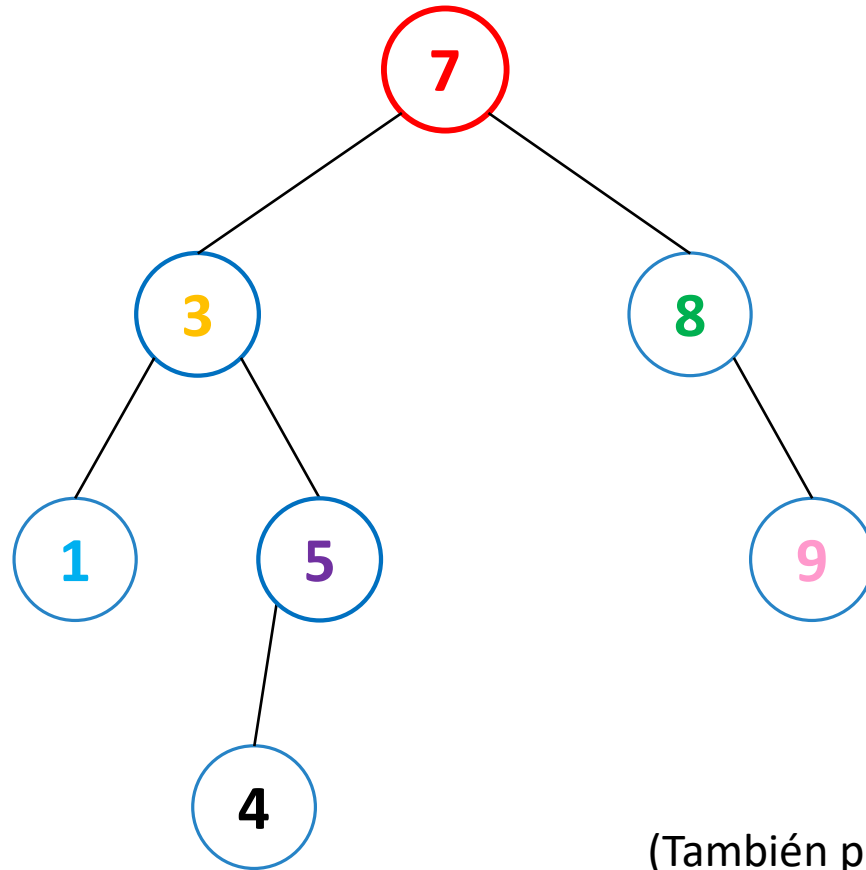
Ahora el 6... no es tan fácil



Se reemplaza por el sucesor



Se remplaza por el sucesor



(También puede ser el antecesor)

Antecesor y Sucesor

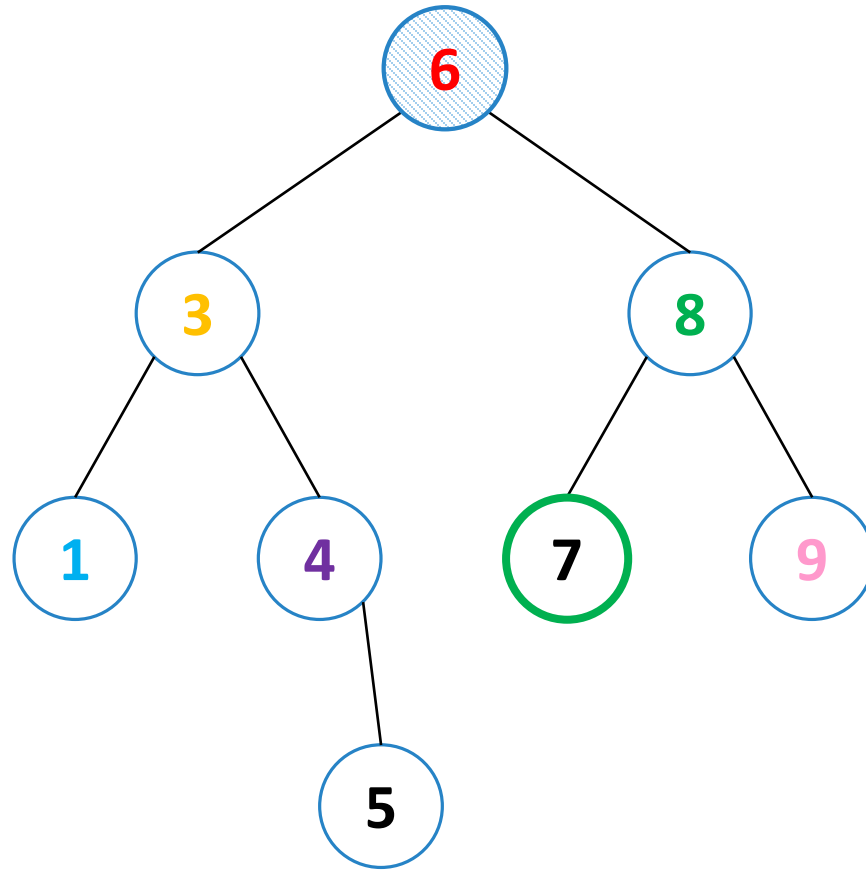


Si los nodos estuvieran ordenados en una lista según su *key*:

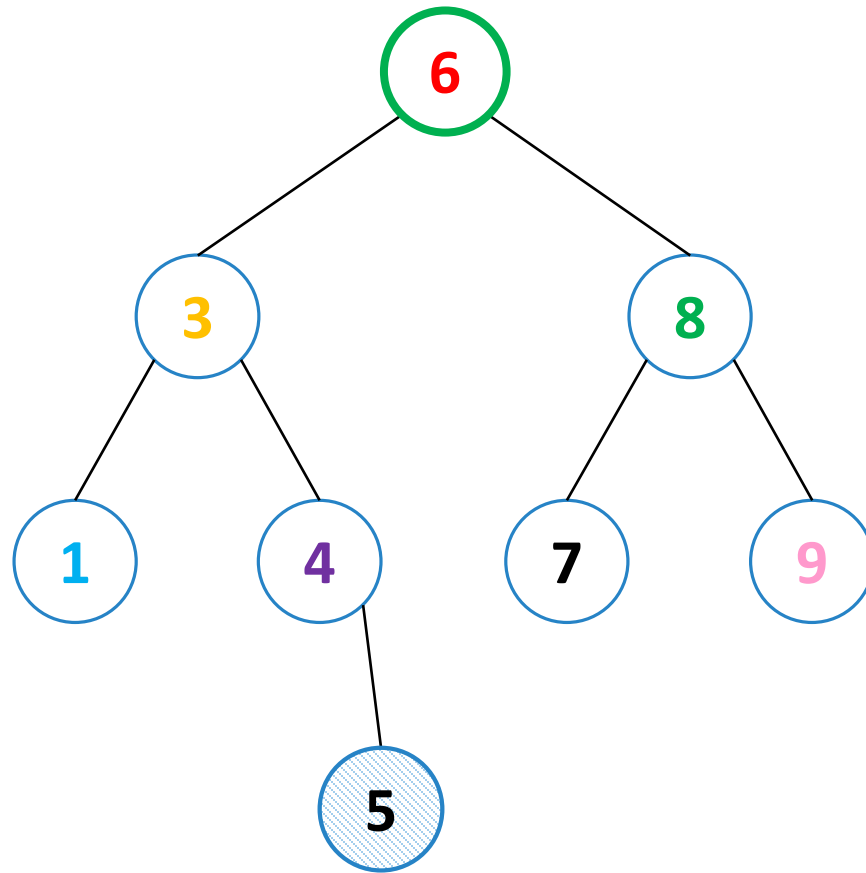
- El **sucesor** de un nodo es el siguiente de la lista
- El **antecesor** de un nodo es el anterior en la lista

¿Cómo podemos encontrar estos elementos dentro del árbol?

Busquemos el sucesor del 6



Busquemos el sucesor del 5



Ya no es tan sencillo... pero no lo necesitamos para eliminar

min(*A*):

if *A.left* = \emptyset :

return *A*

else:

return *min*(*A.left*)

successor(*A*):

if *A.right* $\neq \emptyset$:

return min(*A.right*)

return \emptyset

delete(A, k):

$D = \text{search}(A, k)$

if D es hoja:

$D = \emptyset$

else if D tiene un solo hijo H :

$D = H$

else:

$R = \text{sucesor}(D)$

$D = R$

$R = \emptyset$

El bueno, el malo y el feo

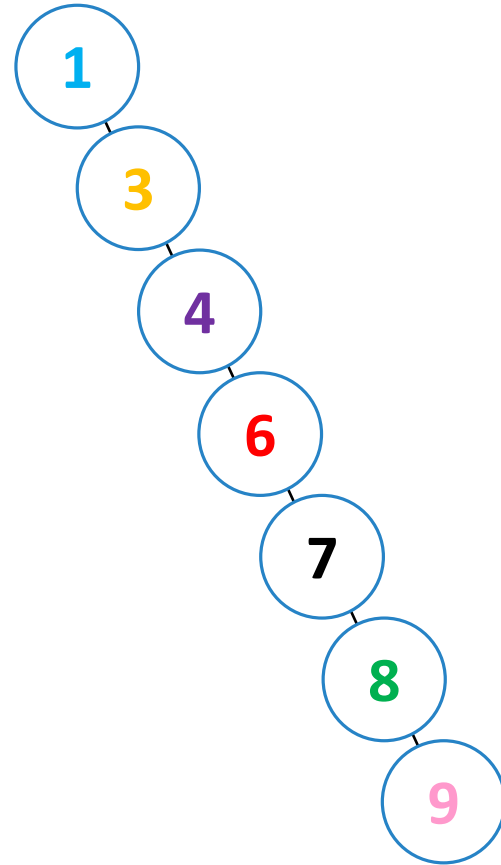
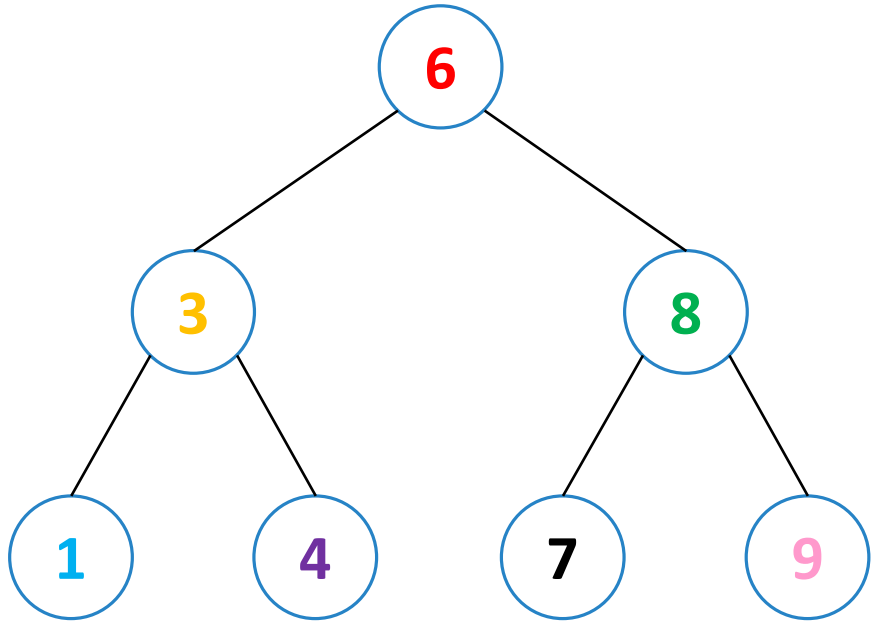


¿Hay algunas raíces más convenientes que otras?

¿Qué pasa con el árbol si no queda un dato conveniente como raíz?

¿Cómo varía la complejidad de las operaciones?

Mismos datos, distinto árbol



Debemos intentar que el árbol sea lo más balanceado posible