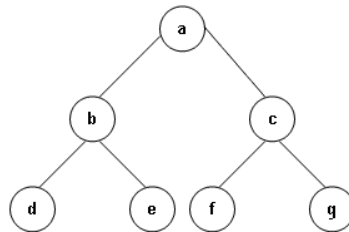


# Árboles binarios

Estructura de Datos

# Árboles binarios

- ▶ A los arboles ordenados de grado dos se les conoce como arboles binarios ya que cada nodo del árbol no tendrá más de dos descendientes directos.
- ▶ Las aplicaciones de los arboles binarios son muy variadas ya que se les puede utilizar para representar una estructura en la cual es posible tomar decisiones con dos opciones en distintos puntos.
- ▶ La representación gráfica de un árbol binario es la siguiente:



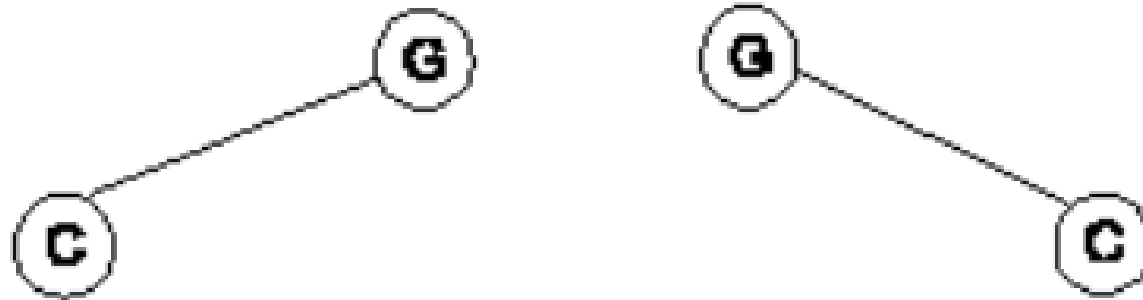
# Clasificación de árboles binarios

▶ Existen cuatro tipos de árbol binario:.

- ▶ A. B. Distinto.
- ▶ A. B. Similares.
- ▶ A. B. Equivalentes.
- ▶ A. B. Completos.

# Árboles Binarios Distintos

- ▶ Se dice que dos árboles binarios son distintos cuando sus estructuras son diferentes



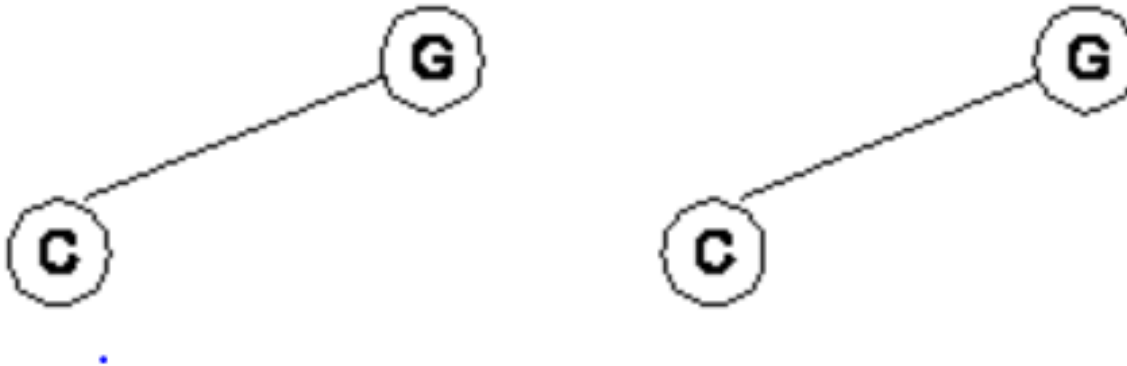
# Árboles Binarios Similares

- ▶ Dos árboles son similares cuando sus estructuras son idénticas, pero la información que contienen sus nodos es diferente:



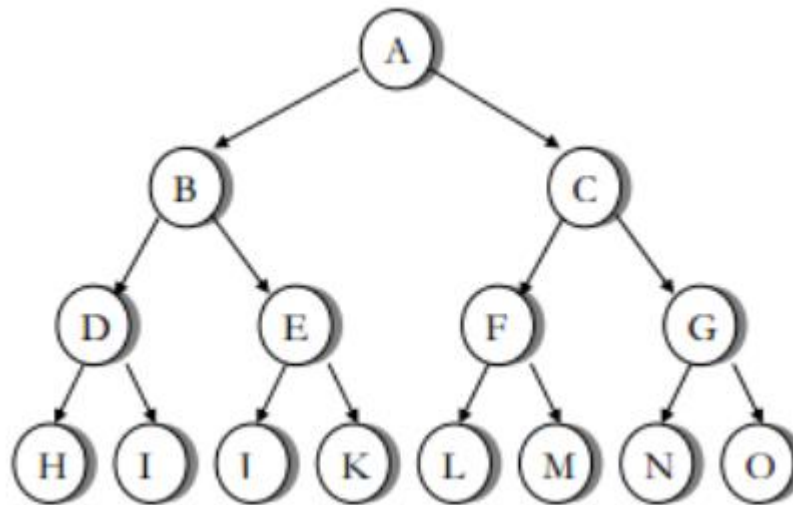
# Árboles Binarios Equivalentes

- ▶ Son aquellos árboles que son similares y que además los nodos contienen la misma información:



# Árboles Binarios Completos

- Son aquellos árboles en los que todos sus nodos excepto los del último nivel, tiene dos hijos, el subárbol izquierdo y el subárbol derecho.



# Árboles binarios de búsqueda

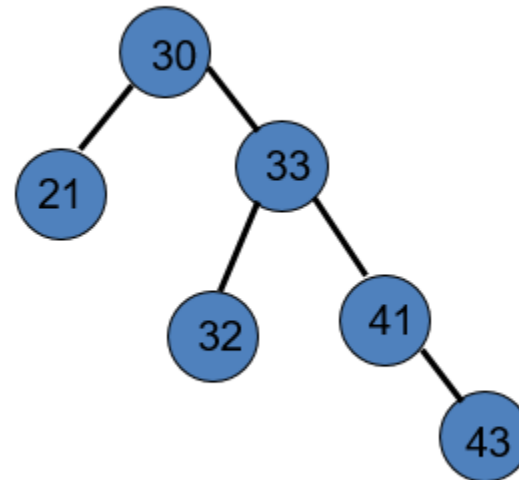
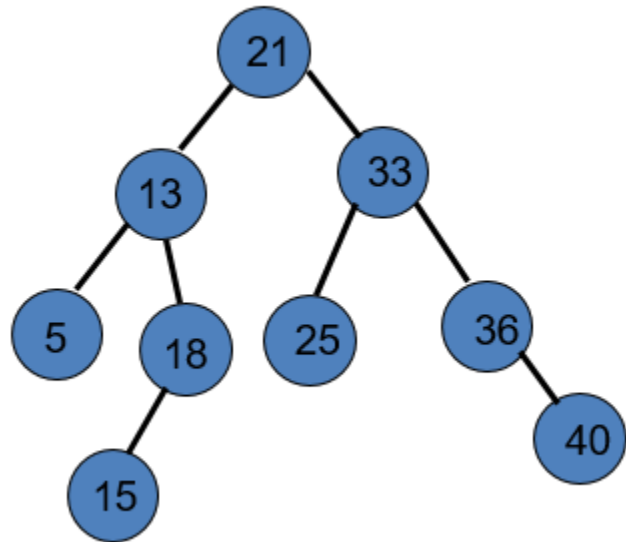
Estructura de Datos



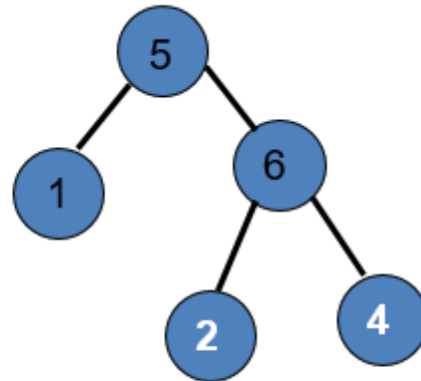
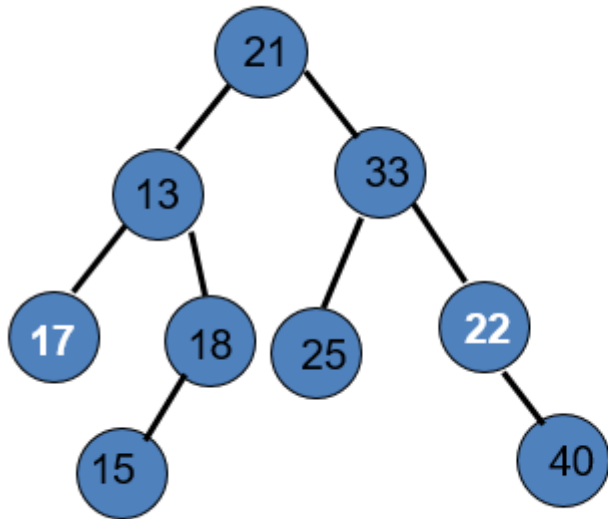
# Árbol Binario de Búsqueda (ABB)

- ▶ Este tipo de árbol permite almacenar información ordenada.
- ▶ Reglas a cumplir:
  - ▶ Cada nodo del árbol puede tener 0, 1 ó 2 hijos.
  - ▶ Los descendientes **izquierdos** deben tener un valor **menor al padre**.
  - ▶ Los descendientes **derechos** deben tener un valor **mayor al padre**.
- ▶ Cada elemento se almacena una sola vez por lo que no existen elementos repetidos.

# Ejemplo de ABB

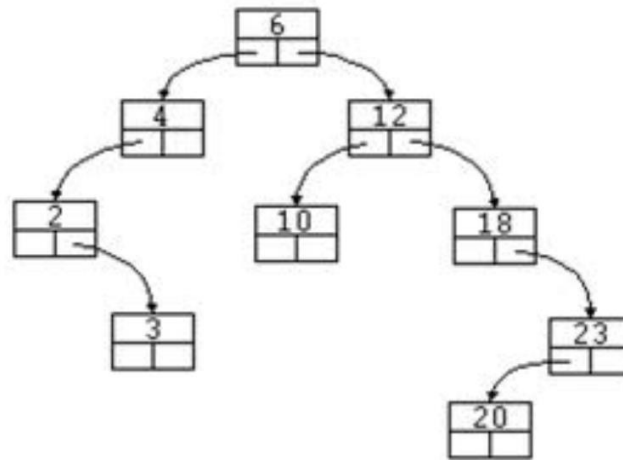


# Ejemplo de no ABB



# Campos

- ▶ Cada elemento(nodo) de un árbol ABB cuenta con tres campos:
  - ▶ Dato(numero, letra, palabra, etc), en este caso usaremos un numero(entero).
  - ▶ Puntero al nodo derecho
  - ▶ Puntero al nodo izquierdo

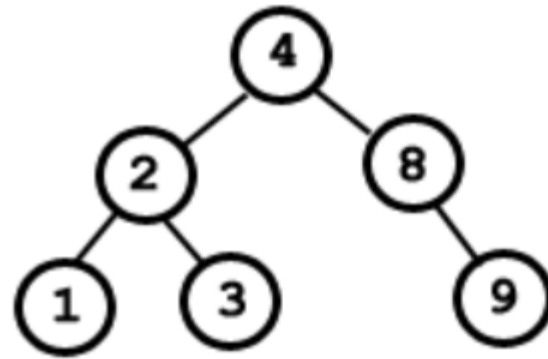


# Inserción

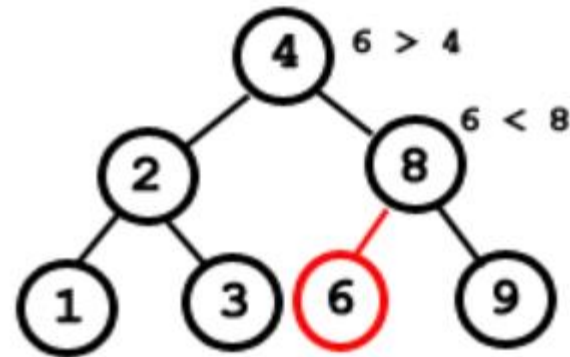
- ▶ Cuando se inserta un nuevo nodo en el árbol hay que tener en cuenta que cada nodo no puede tener más de dos hijos, por esta razón si un nodo ya tiene 2 hijos, el nuevo nodo nunca se podrá insertar como su hijo. Con esta restricción se debe mantener la estructura del árbol.
- ▶ Para localizar el lugar adecuado del árbol donde insertar el nuevo nodo se realizan comparaciones entre los nodos del árbol y el elemento a insertar. El primer nodo que se compara es la raíz, si el nuevo nodo es menor que la raíz, la búsqueda prosigue por el nodo izquierdo de éste. Si el nuevo nodo fuese mayor, la búsqueda seguiría por el hijo derecho de la raíz.
- ▶ Este procedimiento es recursivo, y su condición de parada es llegar a un nodo que no tenga hijo en la rama por la que la búsqueda debería seguir. En este caso el nuevo nodo se inserta en ese hueco, como su nuevo hijo.

# Ejemplo inserción

- ▶ Insertar el nodo 6



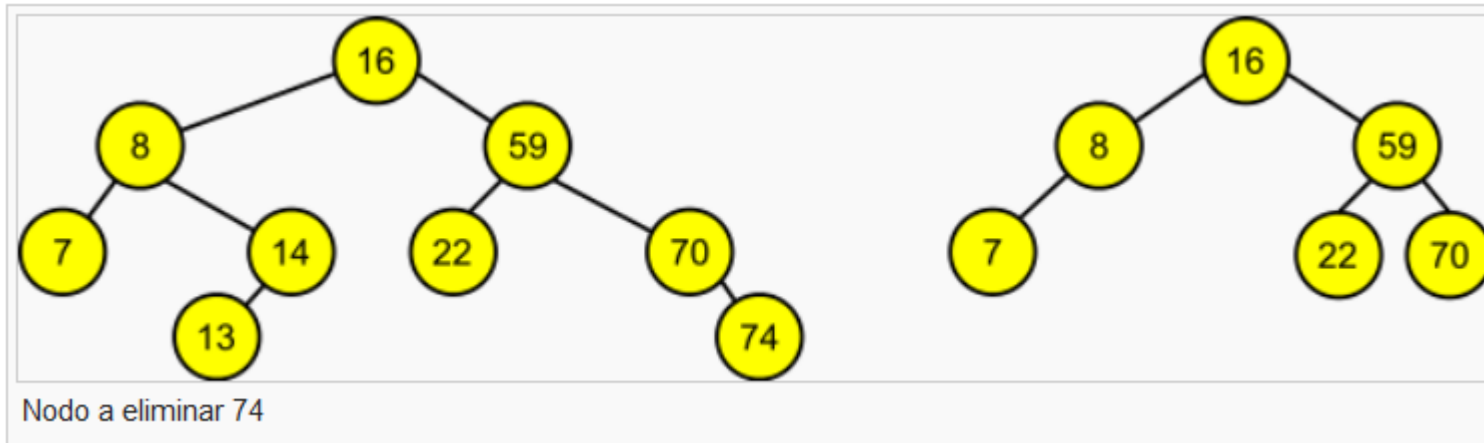
- ▶ Solución:



# Borrar

Borrar un nodo sin hijos o nodo hoja:

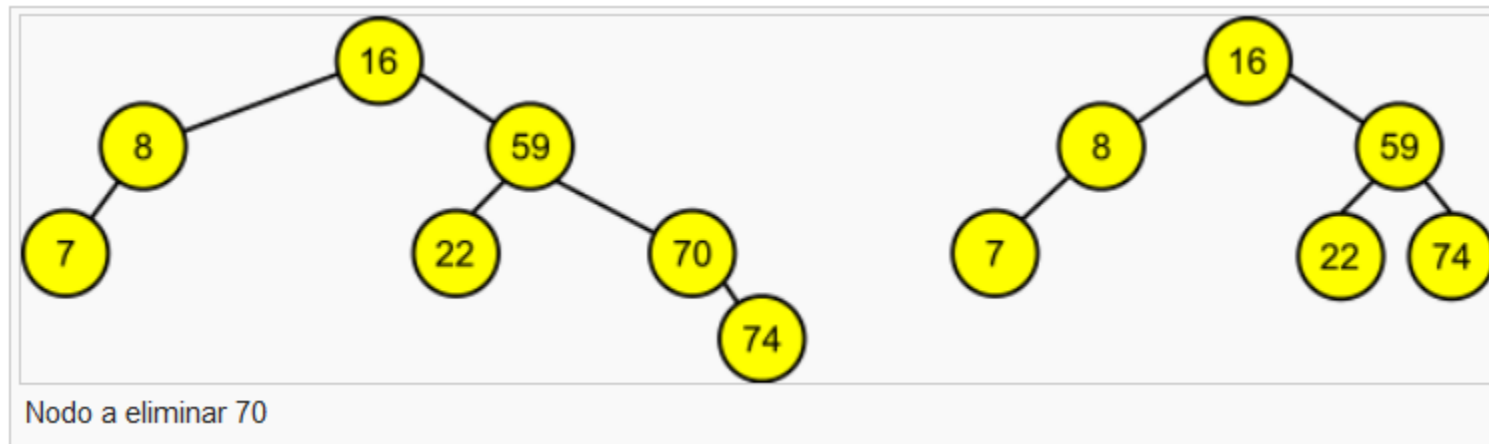
- Simplemente se borra y se establece a nulo el apuntador de su padre.



# Borrar

## Borrar un nodo con un subárbol hijo

- ▶ Se borra el nodo y se asigna su subárbol hijo como subárbol de su padre.

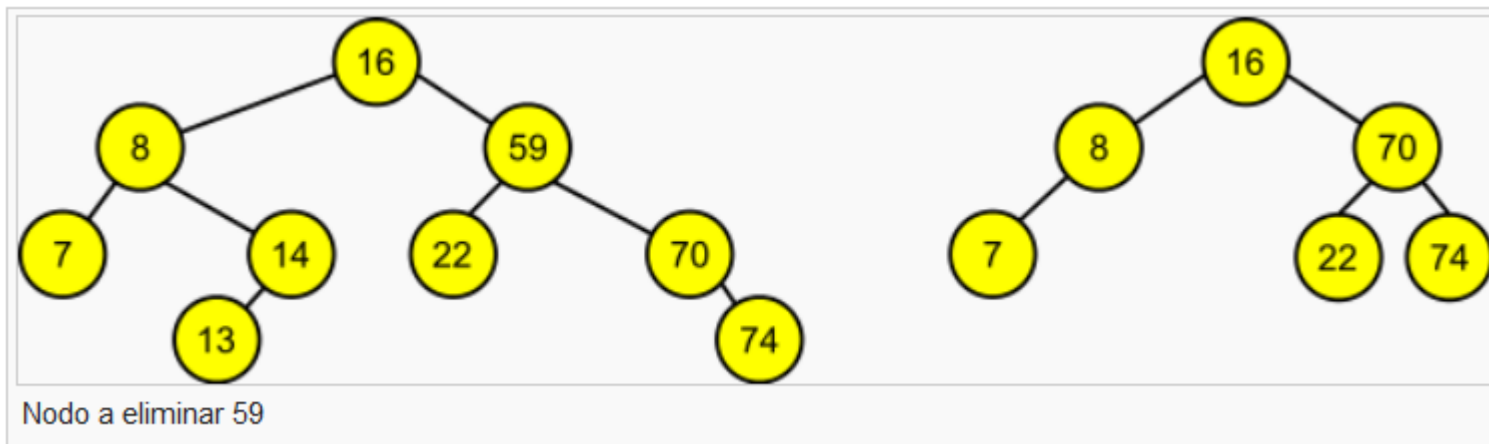




# Borrar

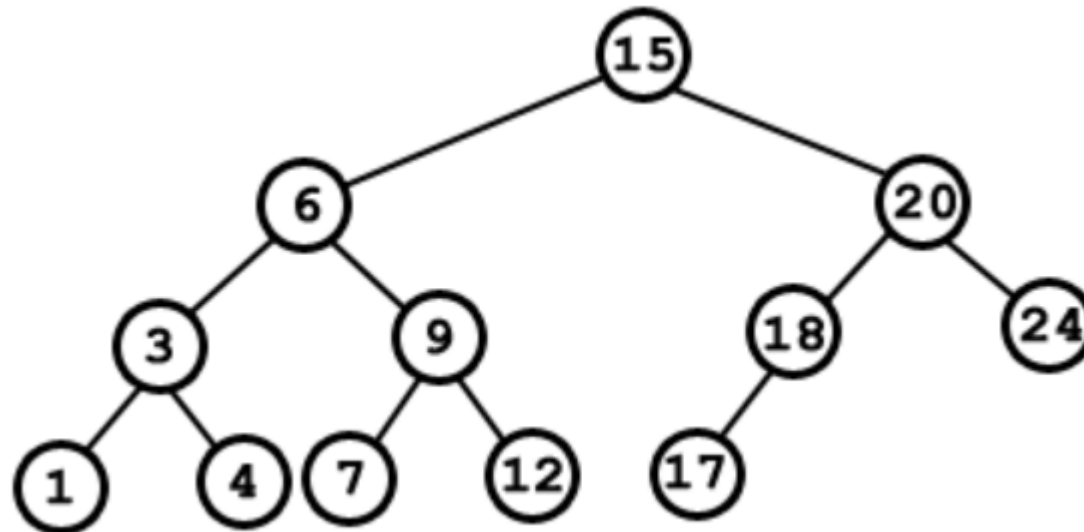
## Borrar un nodo con un subárbol hijo

- ▶ La solución está en reemplazar el valor del nodo por el de su predecesor o por el de su sucesor en inorden y posteriormente borrar este nodo.
- ▶ Su predecesor en inorden será el nodo más a la derecha de su subárbol izquierdo (mayor nodo del subarbol izquierdo), y su sucesor el nodo más a la izquierda de su subárbol derecho (menor nodo del subarbol derecho).
- ▶ En la siguiente figura se muestra cómo existe la posibilidad de realizar cualquiera de ambos reemplazos:



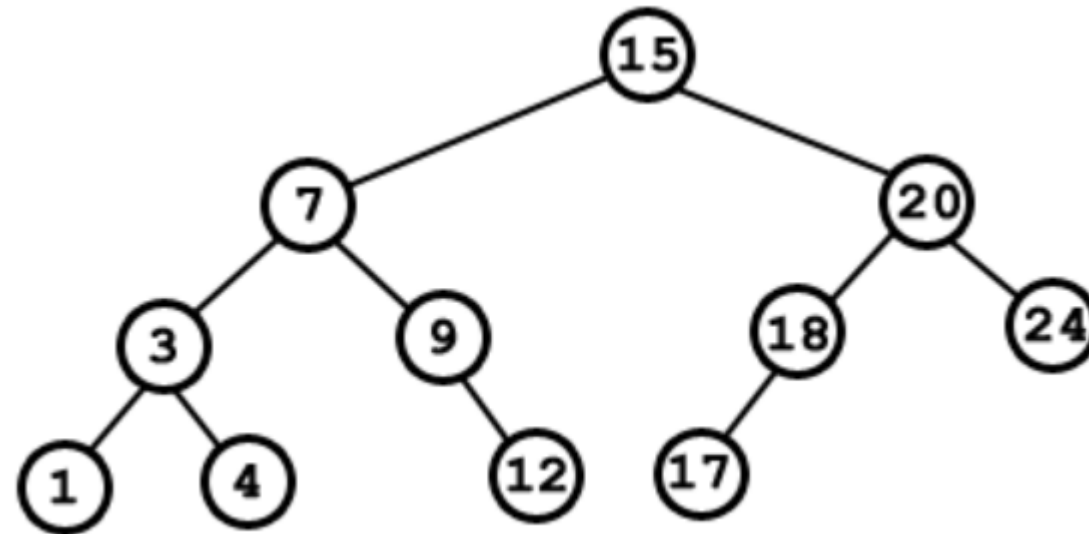
# Ejemplo Borrar

- ▶ Sobre el siguiente árbol queremos eliminar el elemento 6. Tenemos dos opciones para sustituirlo:
- ▶ El menor de sus mayores: 7.
- ▶ El mayor de sus menores: 4.



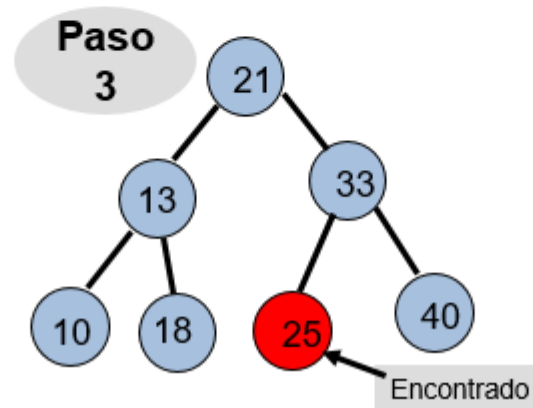
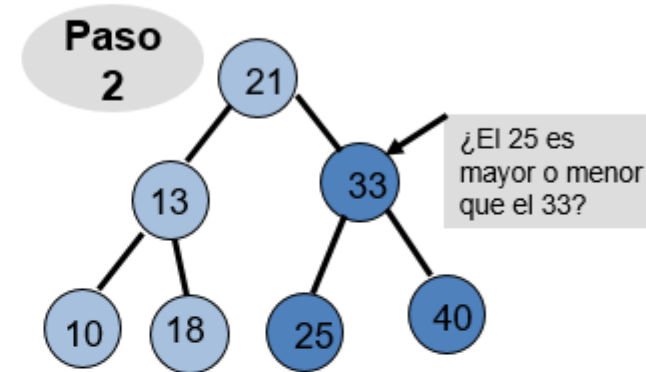
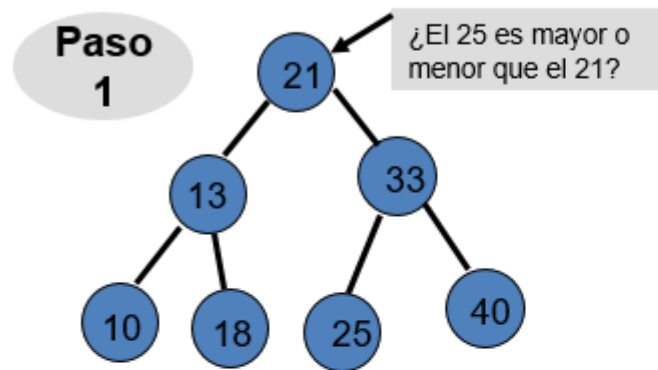
# Ejemplo Borrar

- ▶ Se sustituye el 6 por el 7.
- ▶ El árbol resultante sería el siguiente, tras eliminar también el elemento 7 de su ubicación original.

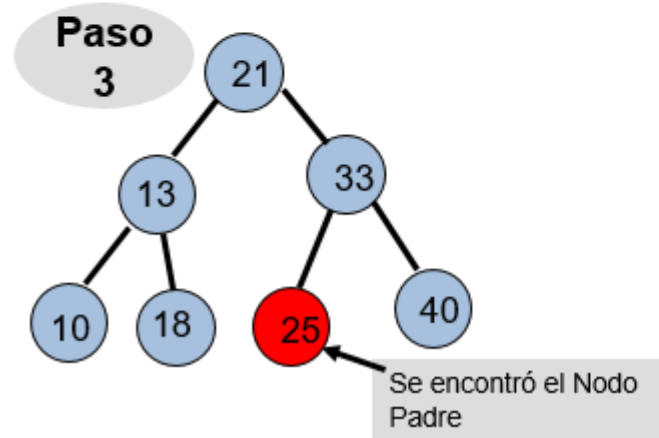
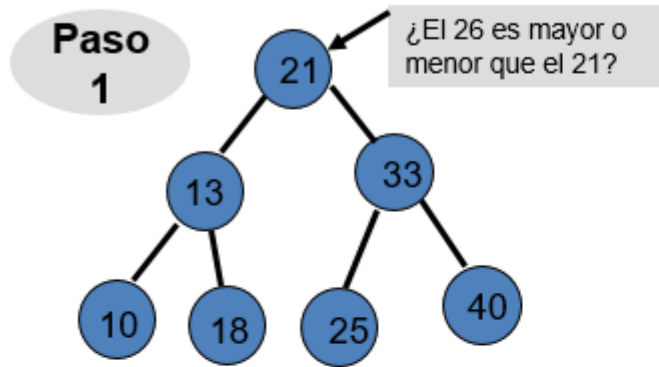


# Proceso para buscar un nodo

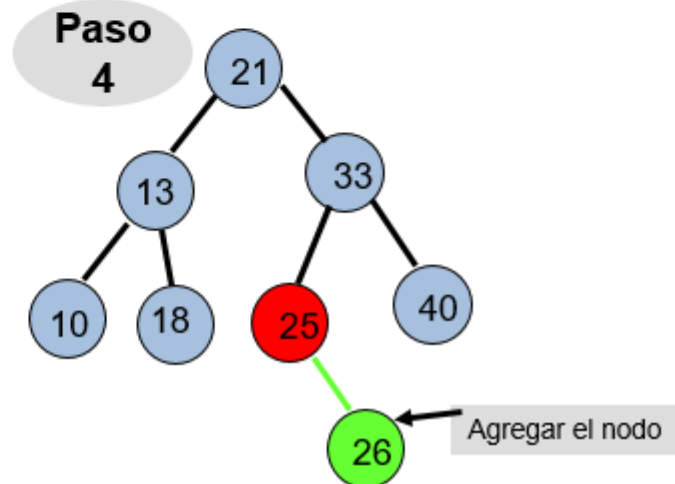
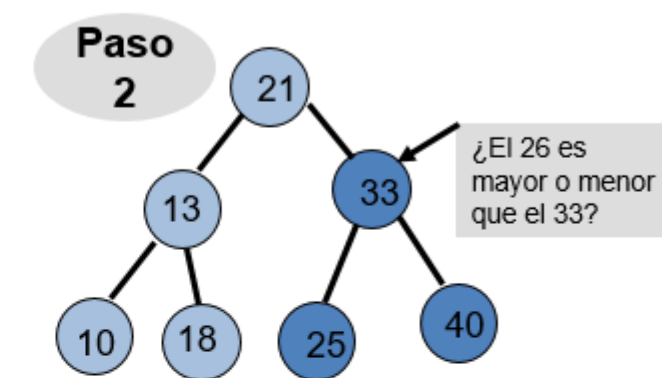
## Buscar el 25



# Ejemplo



## Agregar el valor 26



GRACIAS