

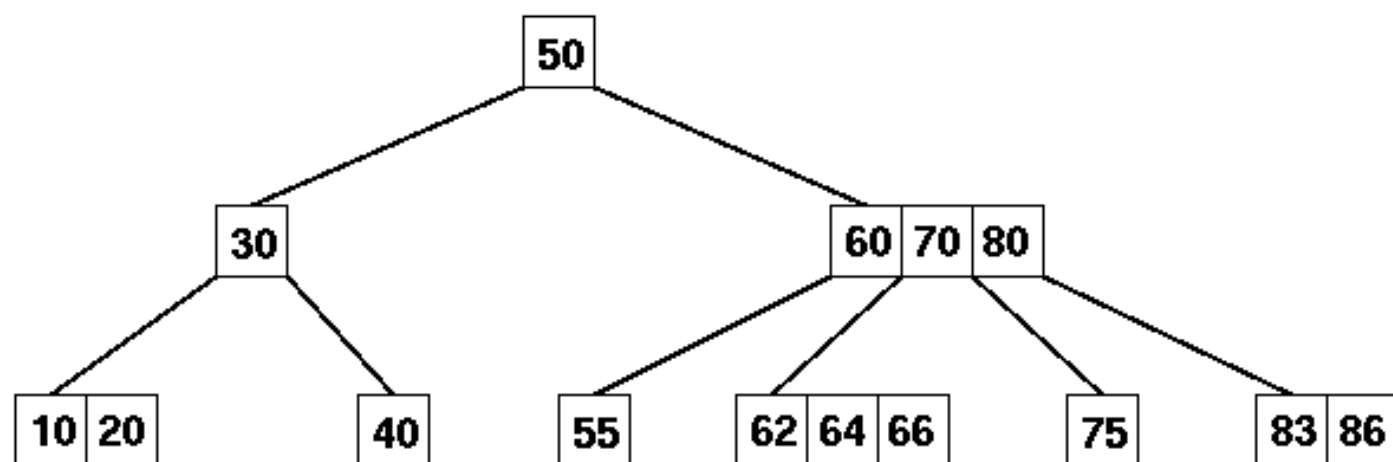
## ÁRBOLES 2-3-4

Los árboles 2-3-4:

- Son 3-arios:
  - Tienen hasta 3 datos por nodo
  - Tienen hasta 4 hijos por nodo
- Todas las hojas están al mismo nivel
- No hay nodos con un solo hijo

Nota: un nodo de  $k$  datos tiene 0 ó  $k+1$  hijos

- Atención: Hay muchas variantes



## ALTURA DE ÁRBOLES 2-3-4

Un árbol 2-3-4 que tenga altura  $h$  tiene al menos tantos nodos como un árbol binario completo de altura  $h$ :

$$n \geq 2^{h+1} - 1$$

por tanto

$$h \leq \log(n+1) - 1$$

Luego un árbol 2-3-4 de  $n$  nodos tiene altura  $\leq \log(n+1) - 1$

**Altura  $O(\log(n))$**

## BÚSQUEDA EN ÁRBOLES 2-3-4

Hay que recorrer el árbol de la raíz a una hoja, luego se tarda tiempo

**$O(\log(n))$**

## INSERCIÓN EN ÁRBOLES 2-3-4

Los nuevos elementos se insertan siempre en las hojas.

Se atraviesa el árbol hacia abajo hasta encontrar la hoja donde insertar.

Dependiendo de si encontramos nodos llenos tenemos **3 casos**:

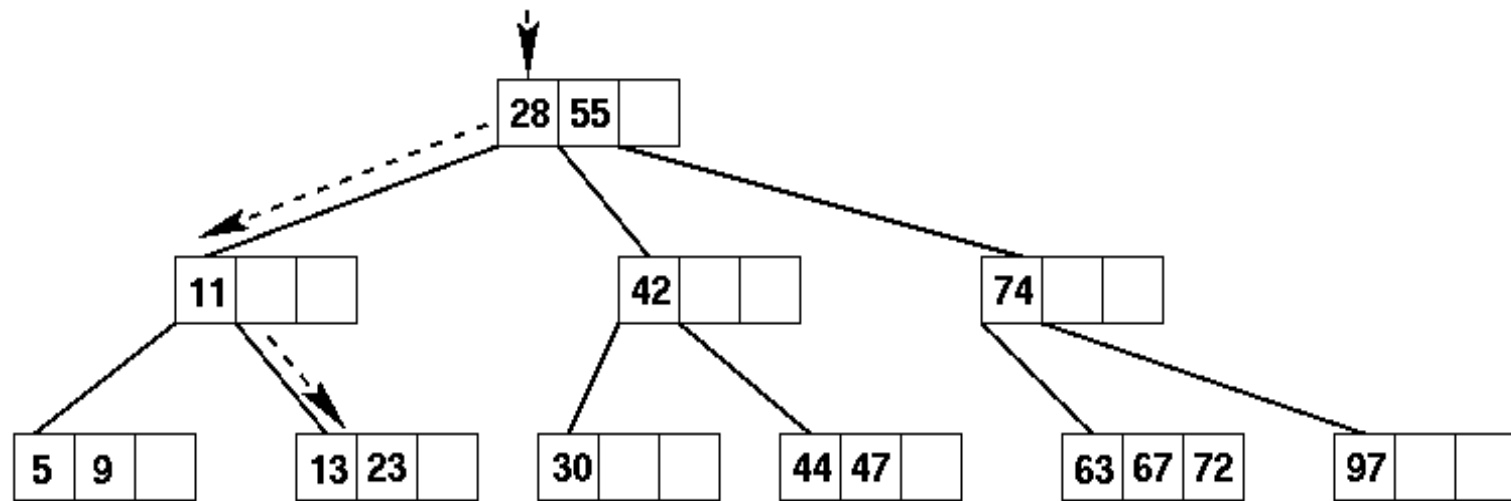
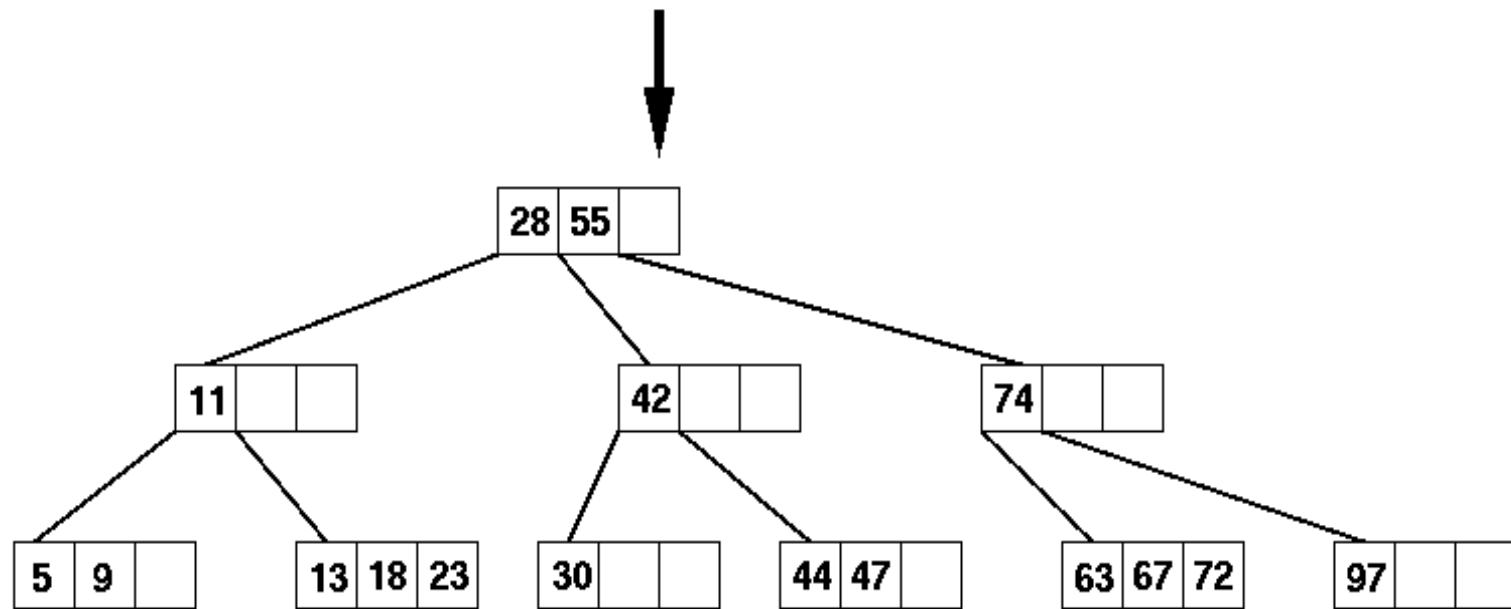
1. inserción sin subdividir nodos
2. inserción subdividiendo nodos
3. inserción subdividiendo la raíz

Queremos insertar de una sola pasada, por tanto cada nodo que encontremos lleno lo subdividimos.

### **Inserción sin subdividir nodos:**

No encontramos nodos llenos en el camino de la raíz a la hoja:

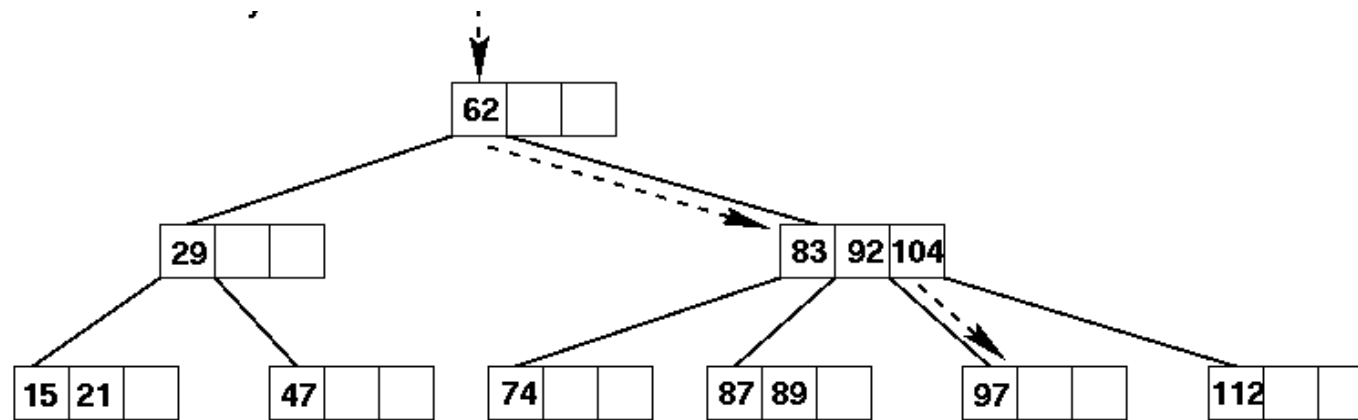
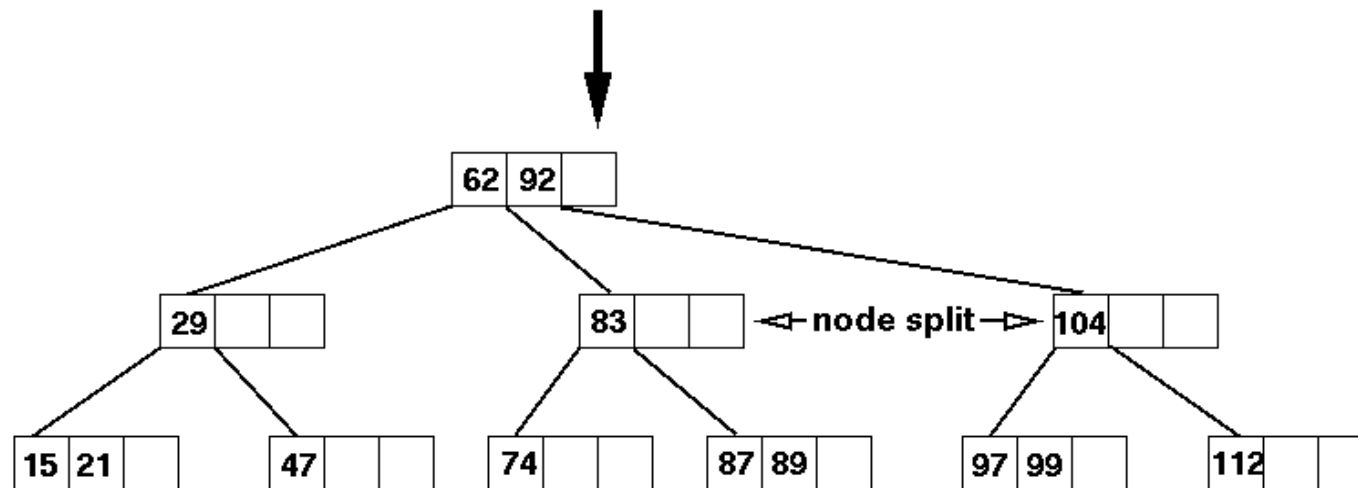
Insertamos 18 en el siguiente árbol.



### **Insertión subdividiendo nodos:**

Cuando encontramos un nodo lleno en el camino de la raíz a la hoja:

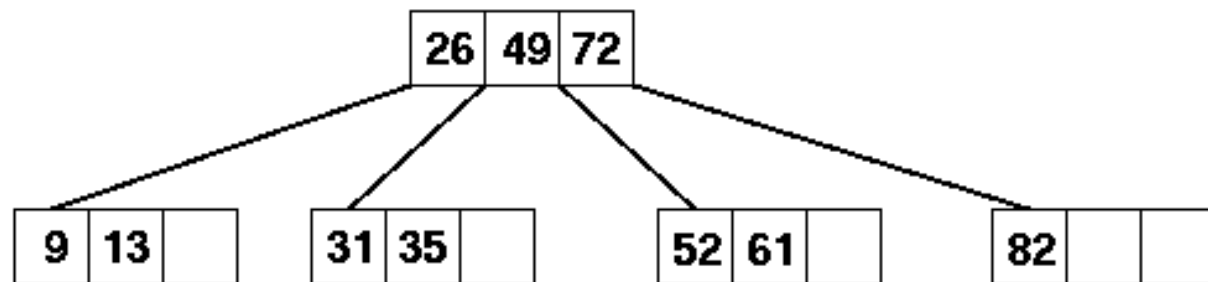
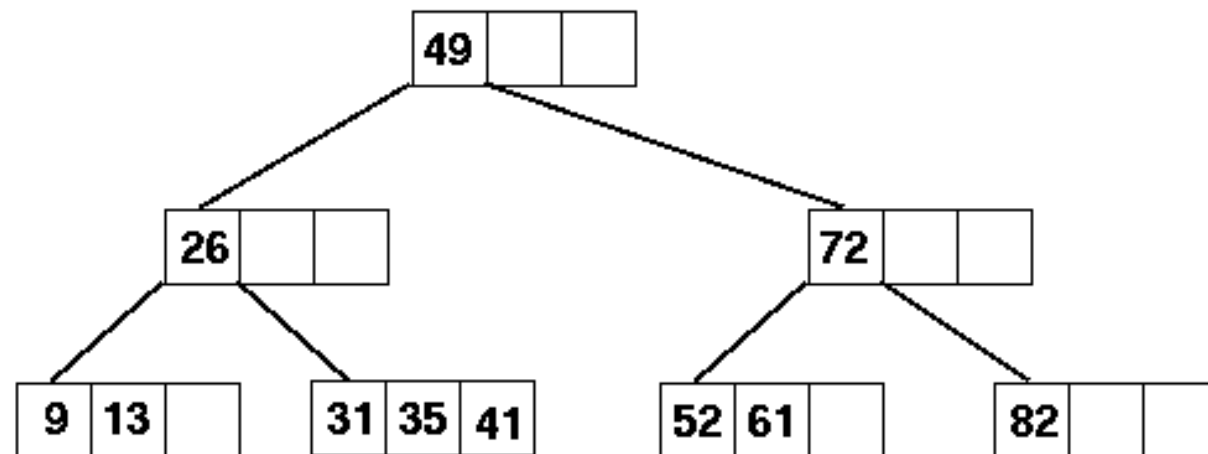
Insertamos 99.



### **Insertión subdividiendo la raíz:**

Esto ocurre cuando nos encontramos la raíz llena. Dividimos la raíz (cuatro hijos) en 3 nodos de dos hijos cada uno.

Insertamos 41.



**Ejemplo:** Generar un árbol 2-3-4 con la secuencia:

70, 50, 30, 40, 20, 80, 25, 90, 75, 10

Resultado:

