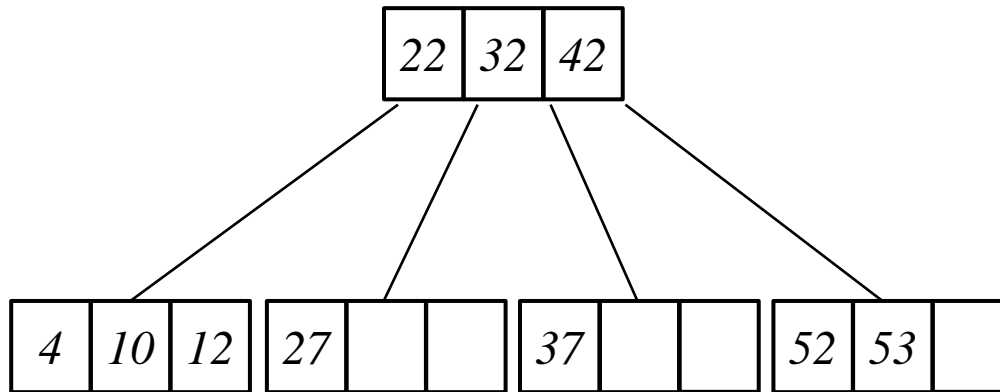


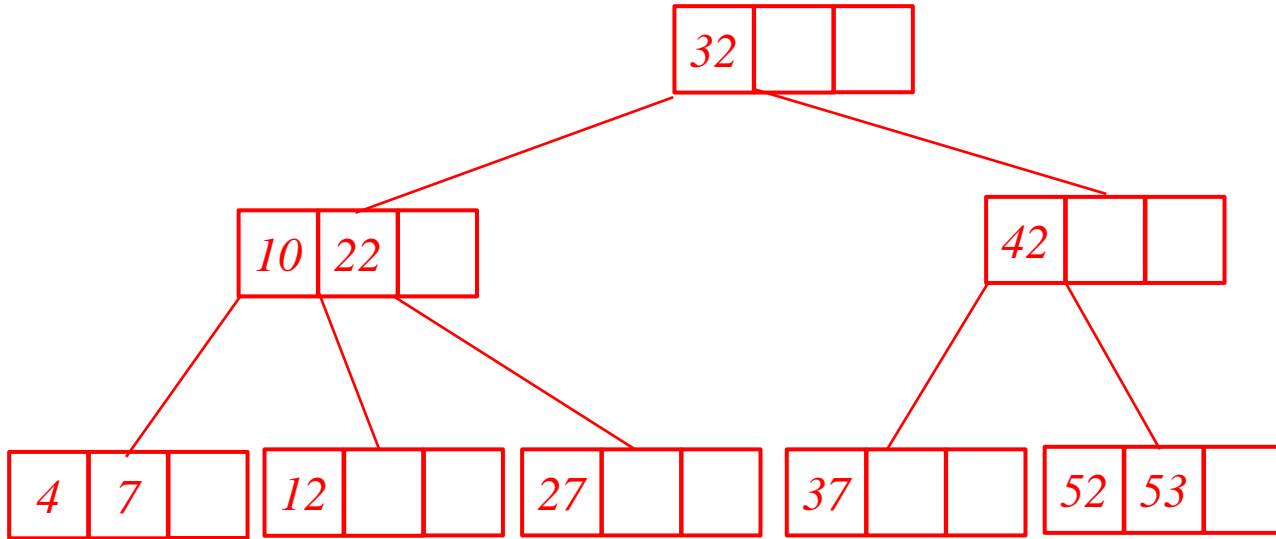
# Más ejemplos de árboles



*árbol B de tres claves por página*

*Insertar la clave 7*

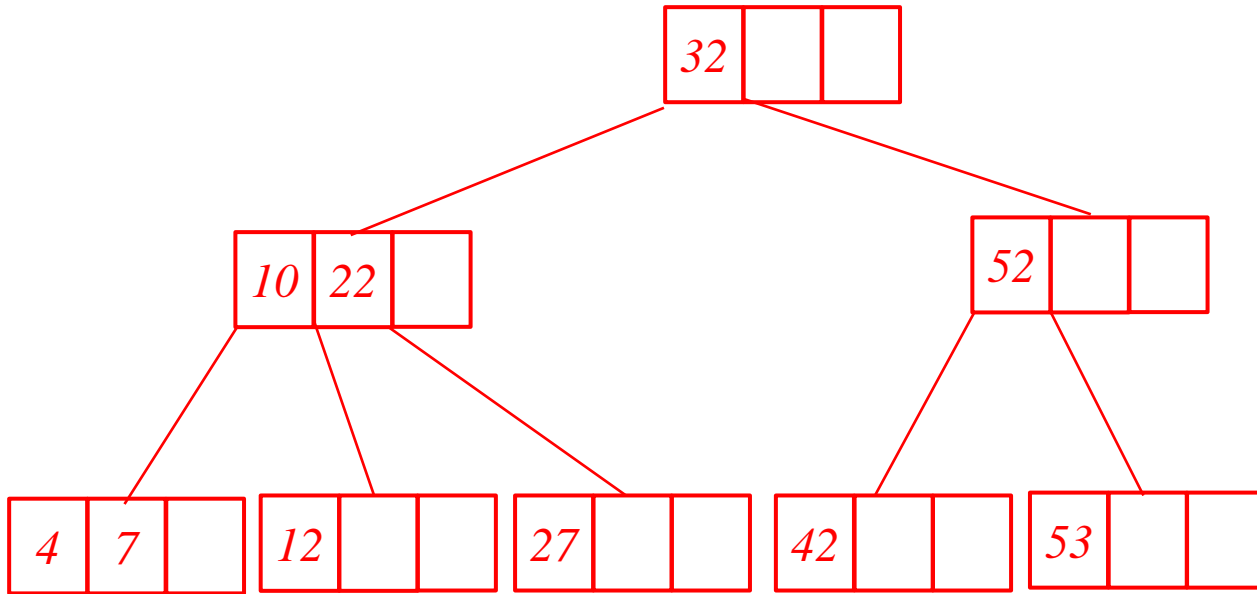
# Más ejemplos de árboles



*árbol B de tres claves por página*

*Eliminar la clave 37*

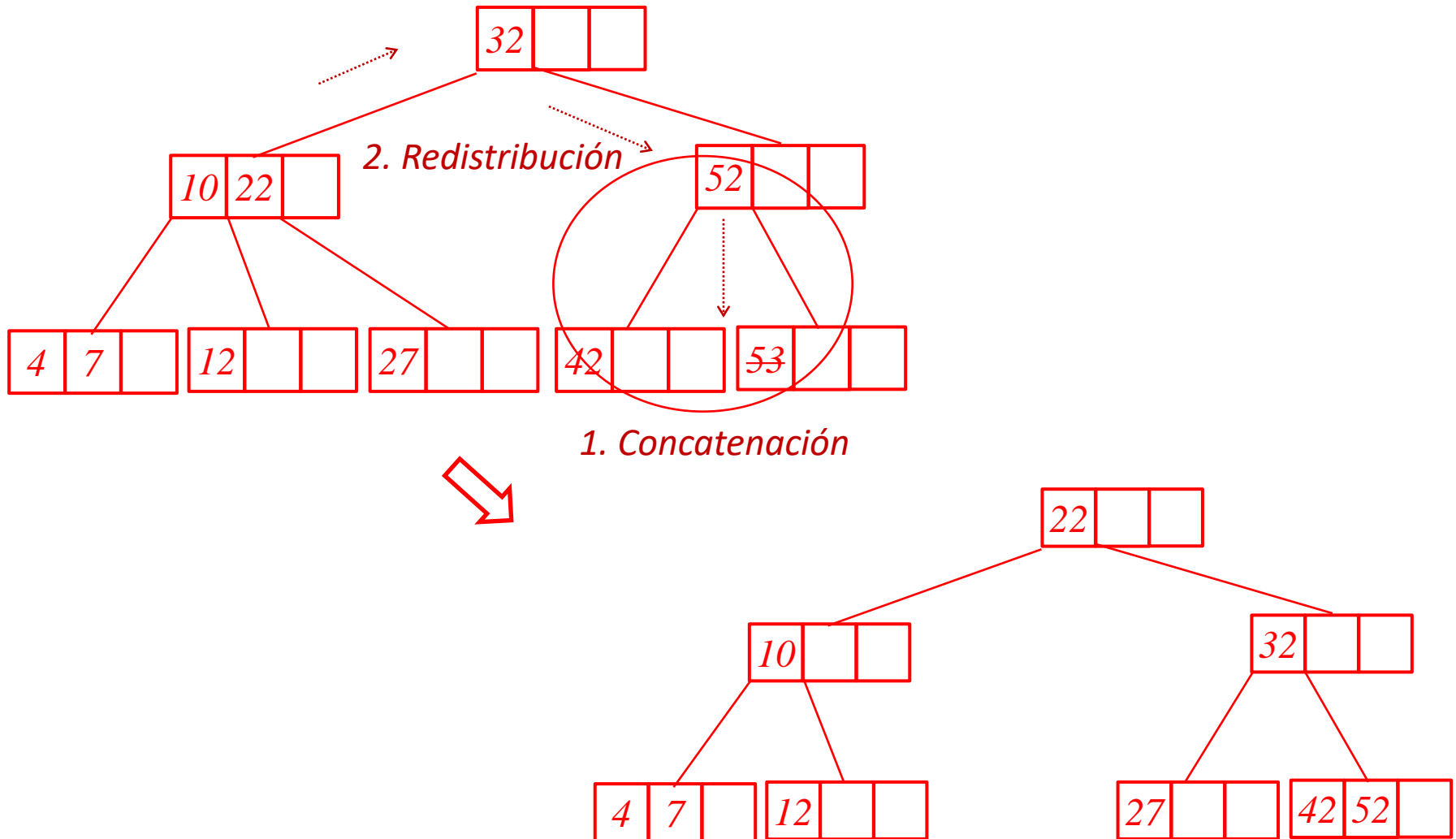
# Más ejemplos de árboles



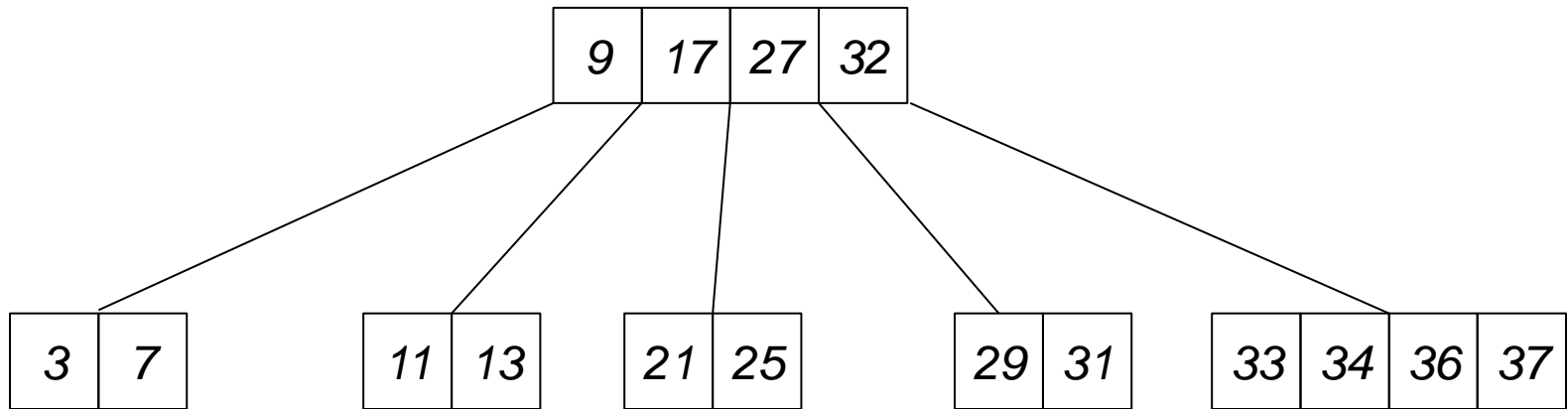
*árbol B de tres claves por página*

*Eliminar la clave 53*

# Más ejemplos de árboles



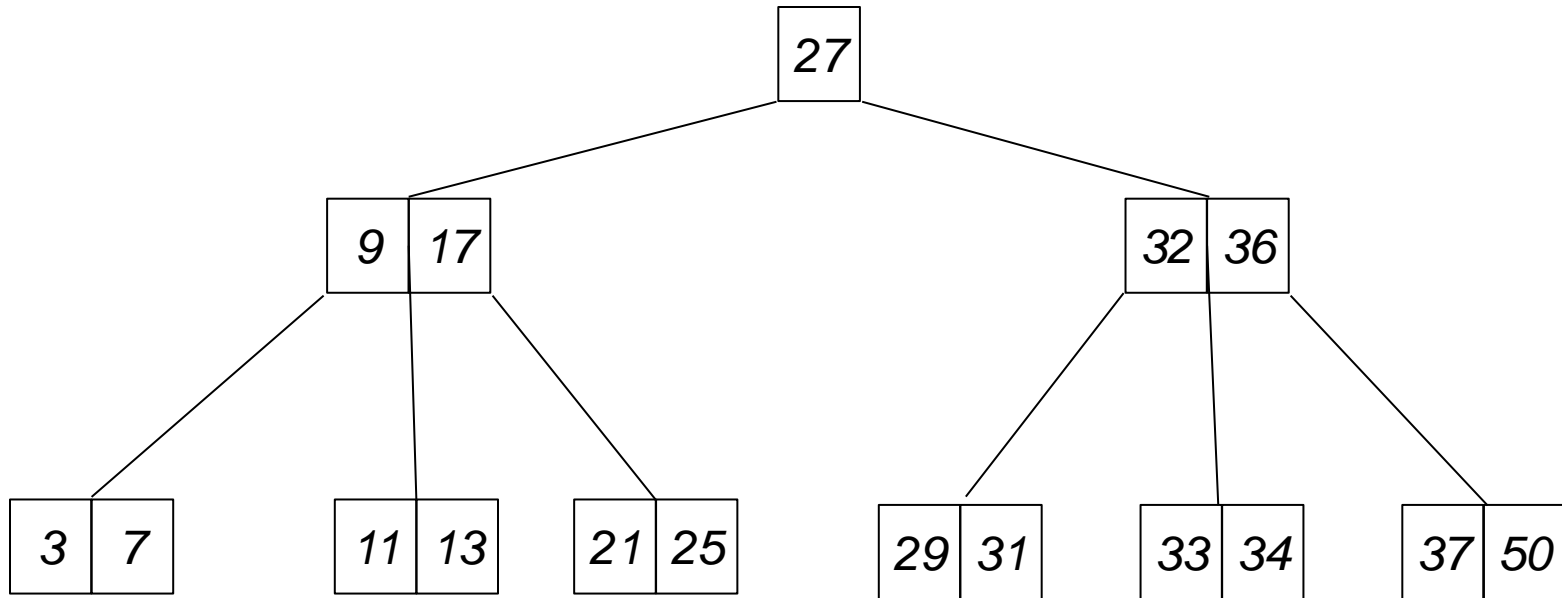
# Más ejemplos de árboles



*árbol B de cuatro claves por página*

*Insertar la clave 50*

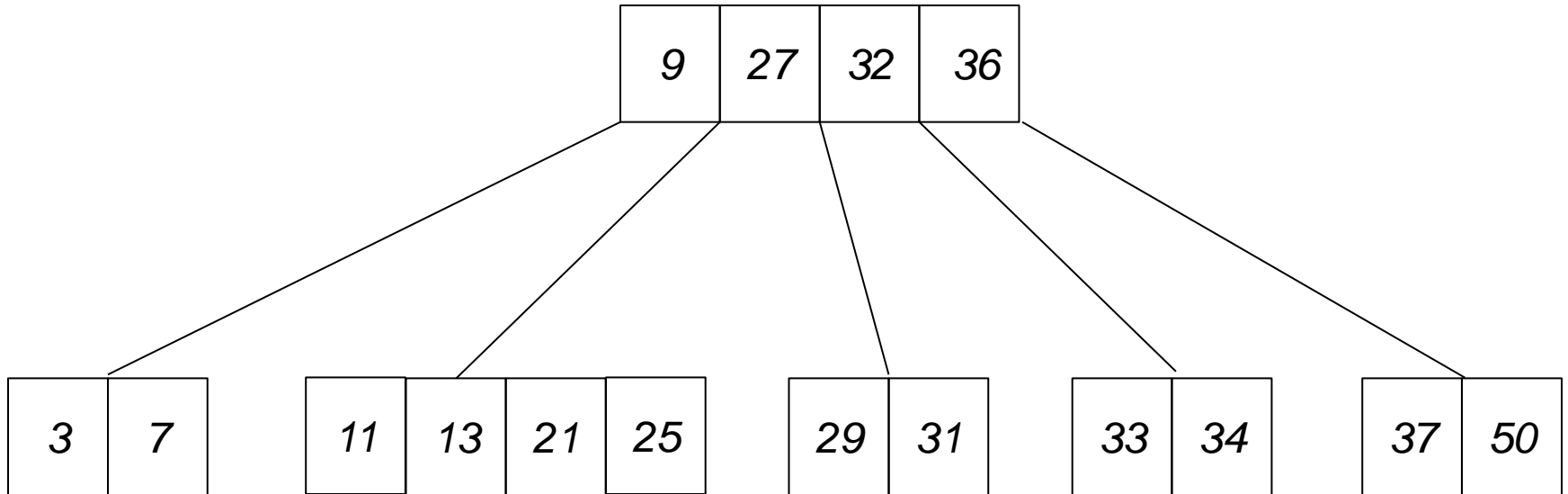
# Más ejemplos de árboles



*árbol B de cuatro claves por página*

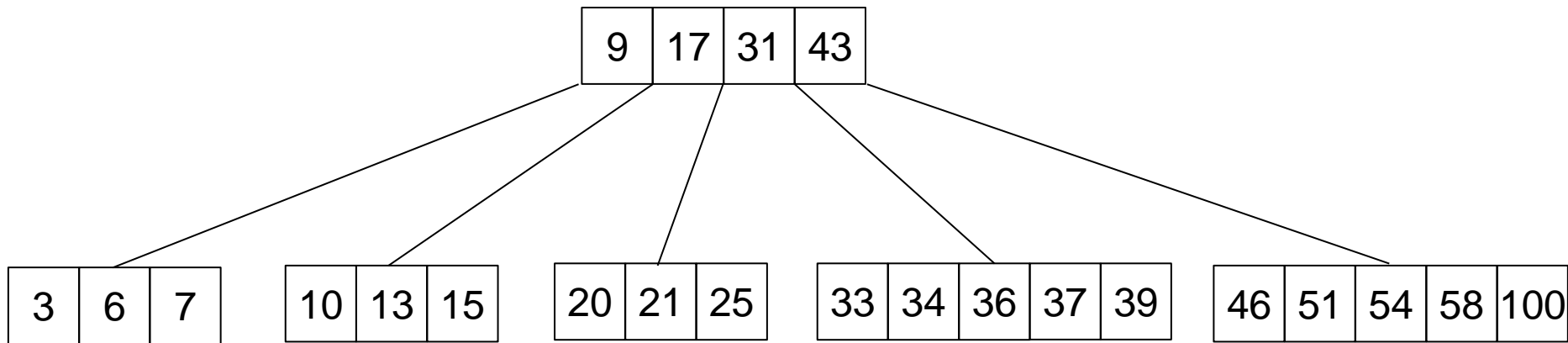
*Eliminar la clave 17*

# Más ejemplos de árboles



*árbol B de cuatro claves por página*

# Más ejemplos de árboles

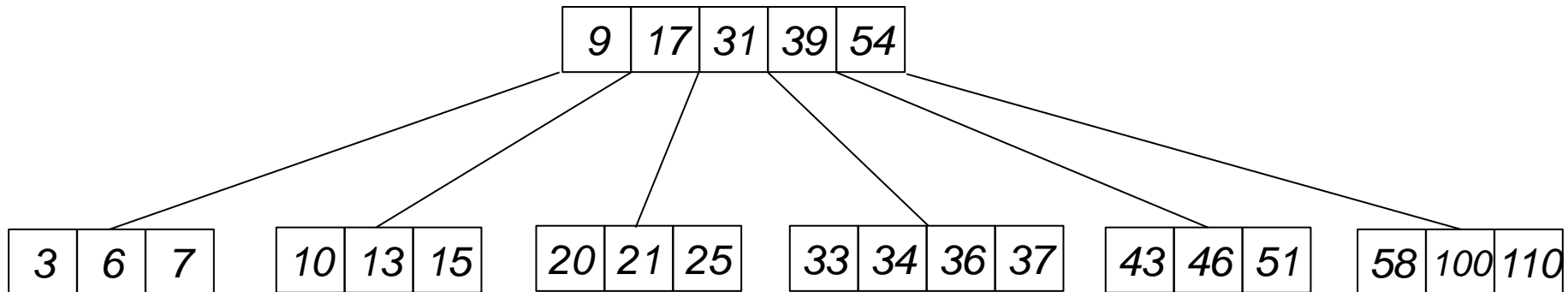
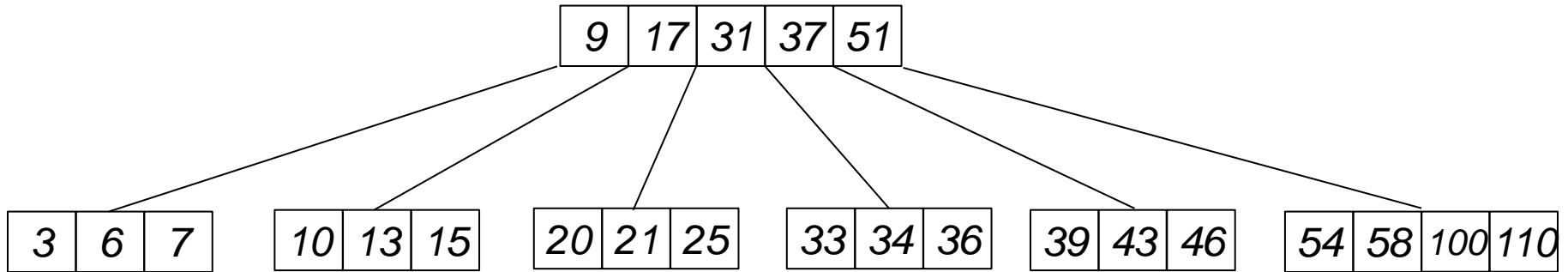


*árbol  $B^*$  de cinco claves por página*

*Insertar la clave 110*



# Más ejemplos de árboles



*árbol  $B^*$  de cinco claves por página*

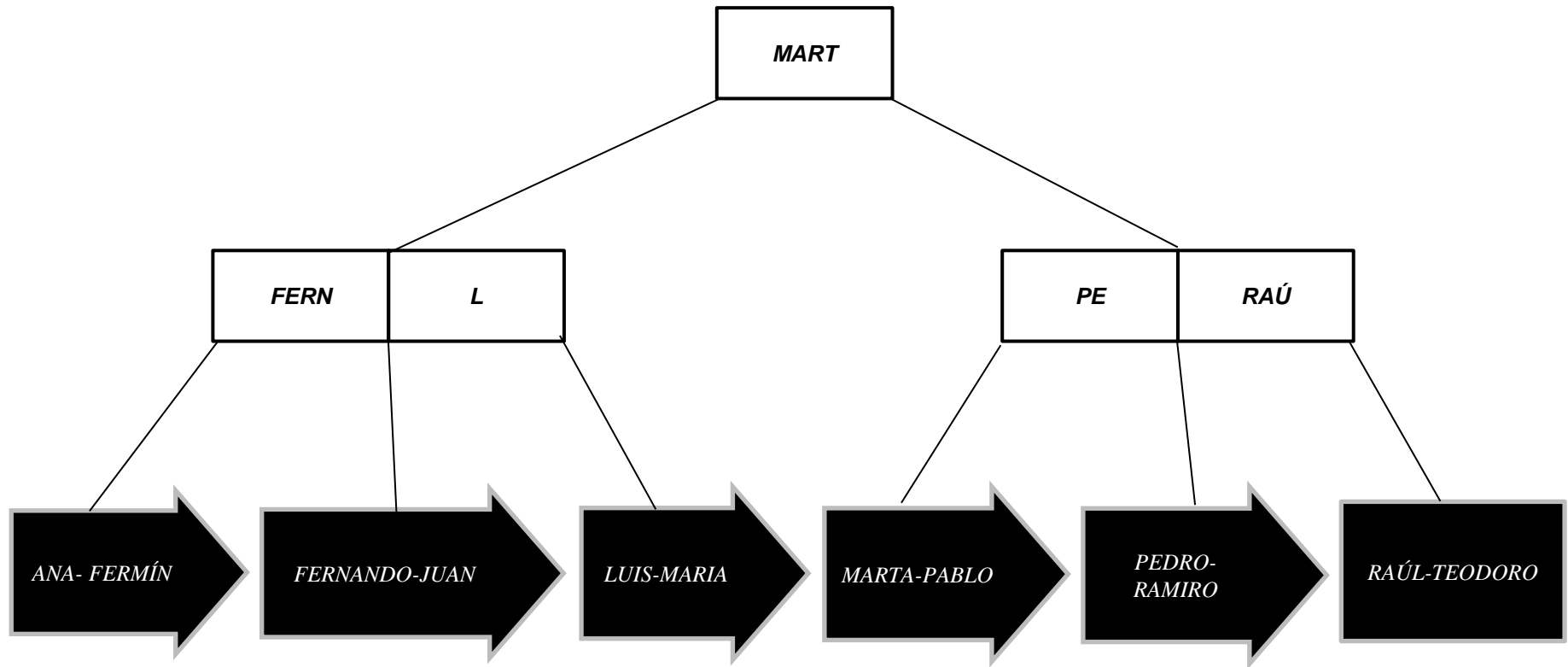
*Dos posibilidades*

# Más ejemplos de árboles



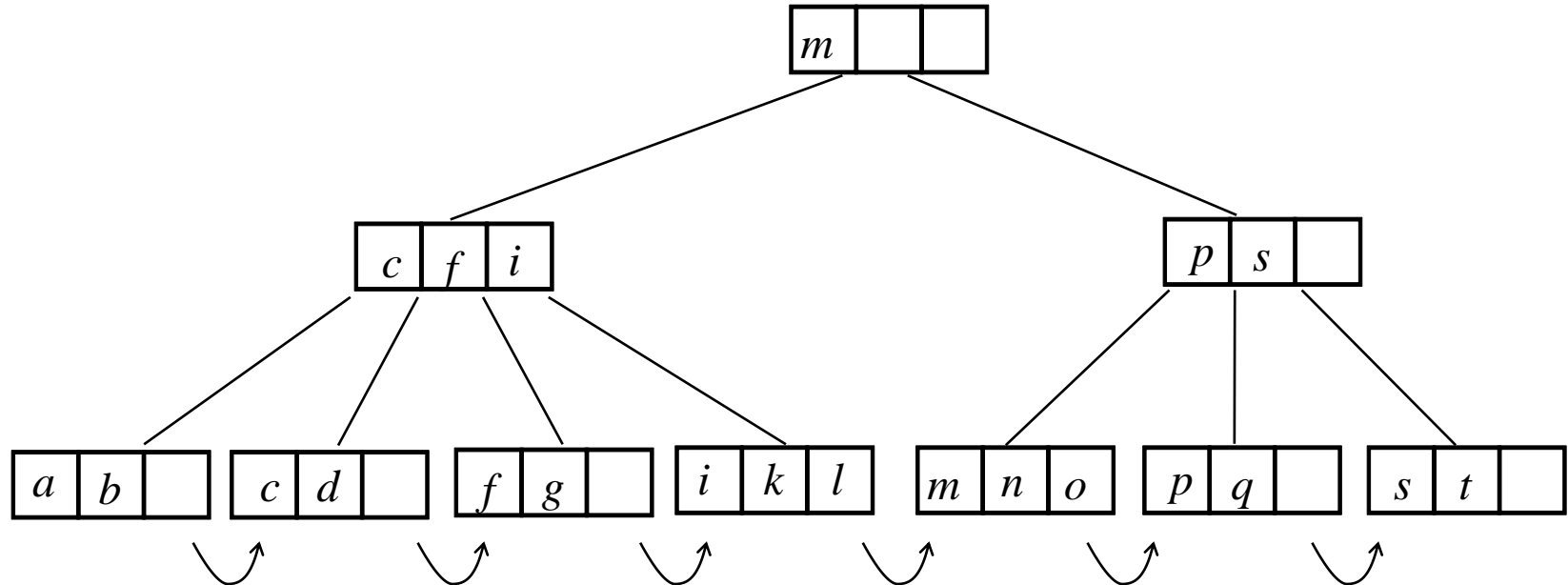
Dados los seis bloques de datos formados cada uno de ellos por varios nombres ordenados, dibujar el árbol de B+ de prefijo simple de 2, 3, 4, 5 claves como máximo por página en el conjunto índice.

# Más ejemplos de árboles



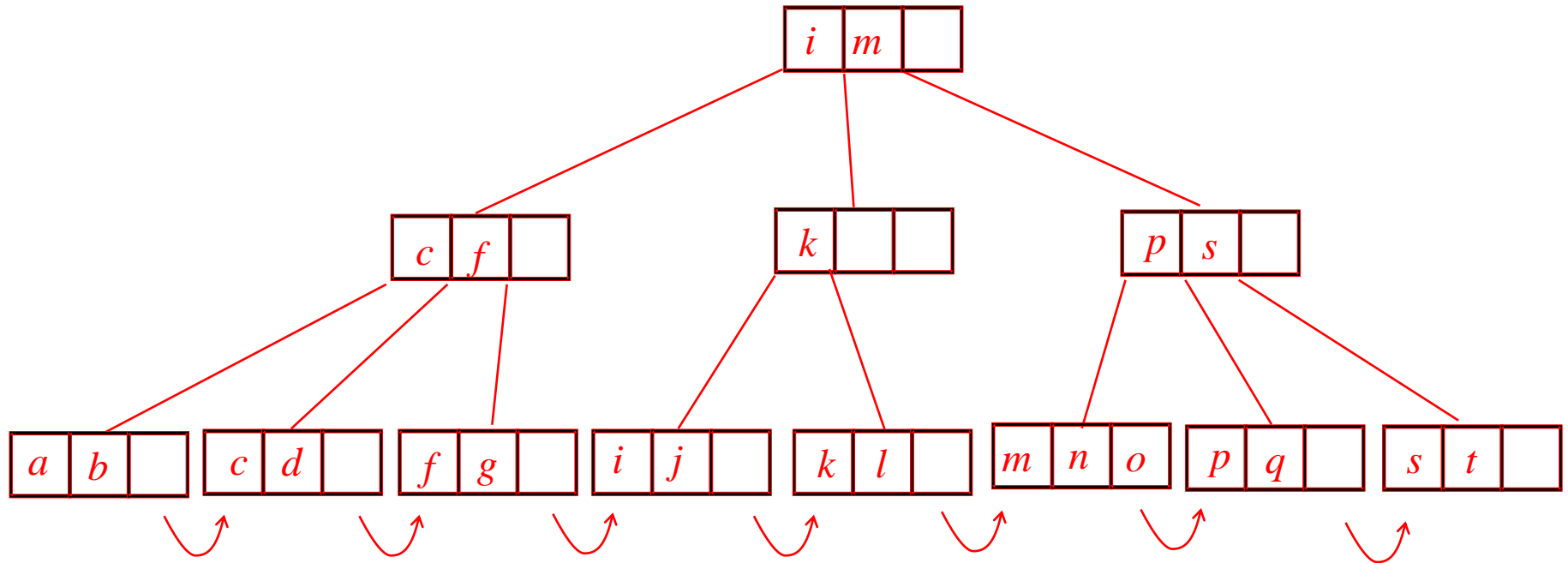
árbol de B+ de prefijo simple de dos claves como máximo por página en el conjunto índice.

# Más ejemplos de árboles



*Árbol B+ de tres claves en el árbol B por página y tres claves en las páginas de la capa hoja, insertar j*

# Más ejemplos de árboles



*Árbol B+ de tres claves en el árbol B por página y tres claves en las páginas de la capa hoja (no hay redistribución)*

# Ejemplo costes de acceso en disco

- ◆ Tenemos un fichero con 160.000 registros de longitud fija de 512 Bytes cada uno en un disco magnético con las siguientes características:
  - 80 sectores por pista, 8 sectores por cluster/bloque, 1024 Bytes por sector, seek promedio de 12 ms y velocidad de rotación de 30.000 rpm.
  - Suponiendo que los clusters/bloques que contienen los registros no están consecutivos y 1Kbyte=1024 Bytes, calcular el tiempo necesario en ms para leer el fichero completo en los supuestos:

Suponiendo que los registros se leen de uno en uno en el programa (sin ningún tipo de buffer de sistema):

Así en el primer caso la lectura de cada cluster involucra los siguientes tiempos:

- Posicionamiento en la pista: seek promedio de **12ms**.
- Latencia rotacional: tiempo en dar una vuelta  $60' / 30.000 = 0.002\text{seg} = \mathbf{2ms}$ , así en promedio será  $2\text{ms} / 2 = \mathbf{1ms}$  de latencia rotacional.
- Tiempo de lectura de un cluster/bloque:  $(8/80) \times 2 = \mathbf{0,2ms}$ , es decir un cluster tiene 8 sectores que corresponde a 8/80 vueltas (en una vuelta se recorren 80 sectores). El tiempo en dar una vuelta son 2ms.

Pero ahora hay que leer 160.000 registros de uno en uno, así que a la fuerza hay que leer para cada registro un cluster (aunque en un cluster quepan más registros). La lectura de registros individuales implica hay por cada uno de ellos seek+latencia rotacional+ tiempo de lectura ( $12+1+0.2 = 13.2\text{ms}$ ). Tiempo total de lectura de los 160.000 registros =  $160.000 \times 13.2 \text{ ms} = 2.112.000\text{ms}$ .

# Ejemplo costes de acceso en disco

Suponiendo que se lee el fichero completo en una sola instrucción en el programa:

Como los clusters/bloques no están consecutivos, por cada bloque hay que hacer un seek + latencia rotacional + el tiempo de lectura del mismo ( $12+1+0.2= 13.2\text{ms}$ ).

Cuantos clusters/bloques hay que leer, es decir cuantos clusters/bloques ocupa el fichero completo:

- 16 regs/bloque y 160.000 reg/fichero  $\Rightarrow 160.000/16$  bloques/fichero = 10.000 bloques para todo el fichero.

Por tanto, leer todo el fichero implica que hay que leer 10.000 bloques, el tiempo de lectura total sería  $10.000 \times 13.2 = 132.000\text{ms}$ .

Suponiendo que se lee el fichero completo en una sola instrucción en el programa y que tuviéramos un hipotético tamaño de sector de 9.000KBytes.

Por cada bloque se necesitan 13.2ms de lectura y ahora el número de bloques del fichero con este tamaño de sector sería:

Hay 8 sectores por cluster/bloque,  $9.000\text{KB} \times 8 = 72.000\text{KB}$  por clusters/bloques. Los 160.000 registros de 512Bytes son 81.920.000 Bytes, o en KB  $81.920.000/1024 = 80.000\text{KB}$ . Así el fichero completo (80.000KB) ocupa 2 clusters/bloques (72.000KB), ya que la mínima operación de entrada o salida es un cluster/bloque. Por tanto, el tiempo total de lectura sería:  $2 \times 13.2 = 26.4\text{ms}$ .

# Ejemplo de organización e índices

Supongamos que la tabla Mercancía se almacena en disco con registros de longitud variable con indicador de longitud como sigue:

14|0|Alimentos

16|1|Combustible

11|2|Madera

12|3|Metales

Utilizando listas enlazadas de registros borrados, mostrar el estado del fichero después de realizar (consecutivamente) las siguientes operaciones, con la estrategia first-fit (y sin considerar la concatenación de registros borrados contiguos).



# Ejemplo de organización e índices

A) Borrar los registros ‘Combustible’, ‘Alimentos’ y ‘Madera’ por ese orden

**Offset** Cabecera: Primer borrado -1

00	14 0 Alimentos
14	16 1 Combustible
30	11 2 Madera
41	12 3 Metales

## **Borrar Combustible**

**Offset** Cabecera: Primer borrado 14

00	14 0 Alimentos
14	16 * -1
30	11 2 Madera
41	12 3 Metales

# Ejemplo de organización e índices

A) Borrar los registros ‘Combustible’, ‘Alimentos’ y ‘Madera’ por ese orden

## **Borrar Combustible**

**Offset** Cabecera: Primer borrado 14

**00** 14|0|Alimentos

**14** 16|\*|-1

**30** 11|2|Madera

**41** 12|3|Metales

## **Borrar Alimentos**

**Offset** Cabecera: Primer borrado 00

**00** 14|\*|14

**14** 16|\*|-1

**30** 11|2|Madera

**41** 12|3|Metales

# Ejemplo de organización e índices

A) Borrar los registros ‘Combustible’, ‘Alimentos’ y ‘Madera’ por ese orden

## Borrar Alimentos

**Offset** Cabecera: Primer borrado 00

00	14 * 14
14	16 * -1
30	11 2 Madera
41	12 3 Metales

## Borrar Madera

**Offset** Cabecera: Primer borrado 30

00	14 * 14
14	16 * -1
30	11 * 0
41	12 3 Metales

# Ejemplo de organización e índices

B) Sobre el resultado de las operaciones anteriores, insertar los registros “16|4|Transportes” y “15|5|Maquinaria”, por este orden

## **Borrar Madera**

**Offset** Cabecera: Primer borrado 30

00	14 * 14
14	16 * -1
30	11 * 0
41	12 3 Metales

## **Insertar 16|4|Transportes**

**Offset** Cabecera: Primer borrado 30

00	14 * -1
14	16 4 Transportes
30	11 * 0
41	12 3 Metales

# Ejemplo de organización e índices

B) Sobre el resultado de las operaciones anteriores, insertar los registros “16|4|Transportes” y “15|5|Maquinaria”, por este orden

## **Insertar 16|4|Transportes**

**Offset** Cabecera: Primer borrado 30

00      14|\*|-1

14      16|4|Transportes

30      11|\*|0

41      12|3|Metales

## **Insertar 15|5|Maquinaria**

**Offset** Cabecera: Primer borrado 30

00      14|\*|-1

14      16|4|Transportes

30      11|\*|0

41      12|3|Metales

53      15|5|Maquinaria

# Ejemplo de organización e índices

Supongamos que la tabla País de una Base de Datos se almacena en un fichero con registros de longitud variable como sigue, utilizando un indicador de longitud (2 bytes) al principio de cada registro, y el separador '|' entre campos:

24|3|Marruecos|Africa|36

21|1|España|Europa|47

20|5|USA|América|327

22|2|Francia|Europa|67

23|0|Alemania|Europa|83

17|4|UK|Europa|68

# Ejemplo de organización e índices

A) Muestra con detalle un índice (simple) primario de esta tabla

## Offset

0	24 3 Marruecos Africa 36
24	21 1 España Europa 47
45	20 5 USA América 327
65	22 2 Francia Europa 67
87	23 0 Alemania Europa 83
110	17 4 UK Europa 68

## Clave 1ª Offset

0	87
1	24
2	65
3	0
4	110
5	45

# Ejemplo de organización e índices

B) Muestra con detalle un índice (simple) secundario por continente.

## Offset

0	24 3 Marruecos Africa 36
24	21 1 España Europa 47
45	20 5 USA América 327
65	22 2 Francia Europa 67
87	23 0 Alemania Europa 83
110	17 4 UK Europa 68

## Clave 2º Clave 1º

Africa	3
America	5
Europa	0
Europa	1
Europa	2
Europa	4