

Arbol B⁺

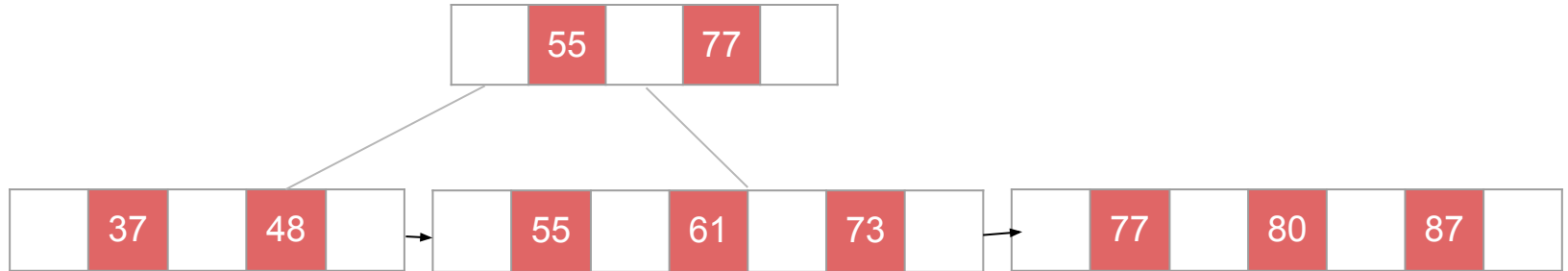
Ing. Max Alejandro Antonio Cerna
Flores

Agenda

- Definición
- Historia
- Características
- Operaciones
 - Búsqueda
 - Inserción
 - Borrado
- Implementaciones de árboles B y B+
- Ventajas y Desventajas

Definición

Son una variante de los árboles B, se diferencian en que los árboles B+ toda la información se encuentra almacenada en las hojas.



Historia

No hay artículo que presente el concepto de árbol B+.

La idea de mantener todos los datos en nodos hoja se menciona repetidamente como una variante interesante de los árboles B.

IBM utilizó árboles B+ por primera vez en 1973 en VSAM (Virtual Storage Access Method).

Características

Maneja un **grado (M)** que indica el número máximo de hijos soportados en un nodo.

La raíz almacena como máximo $M-1$ datos (claves).

Todos los nodos hoja tienen la misma altura.

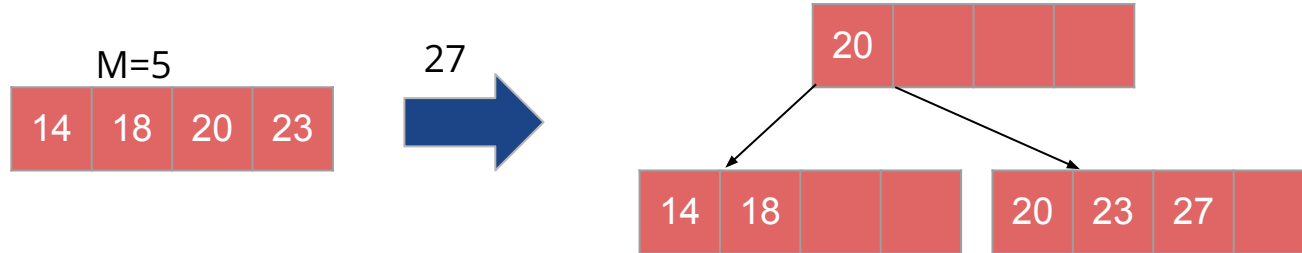
La información se encuentra ordenada.

Los nodos raíz o interiores (no hojas) se utilizan como índices.

Inserción

Similar al árbol B.

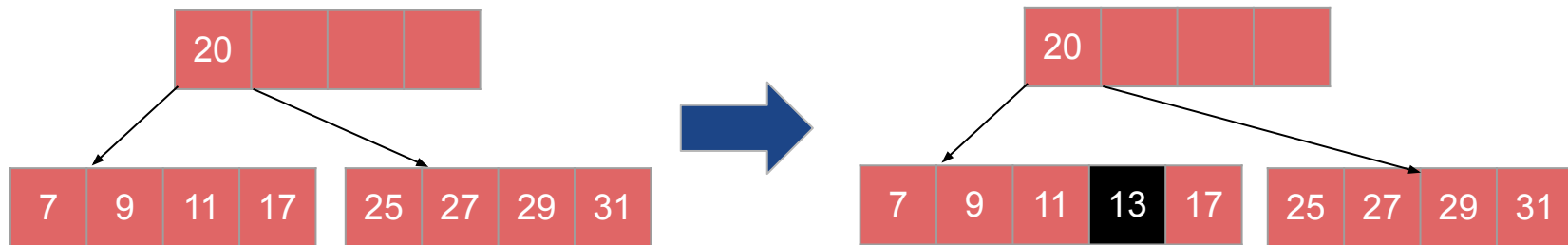
La principal diferencia ocurre cuando se inserta una key en un nodo que se encuentra lleno, se debe hacer una copia de la key que se sube al siguiente nivel en la hoja correspondiente.



Inserción

M=5

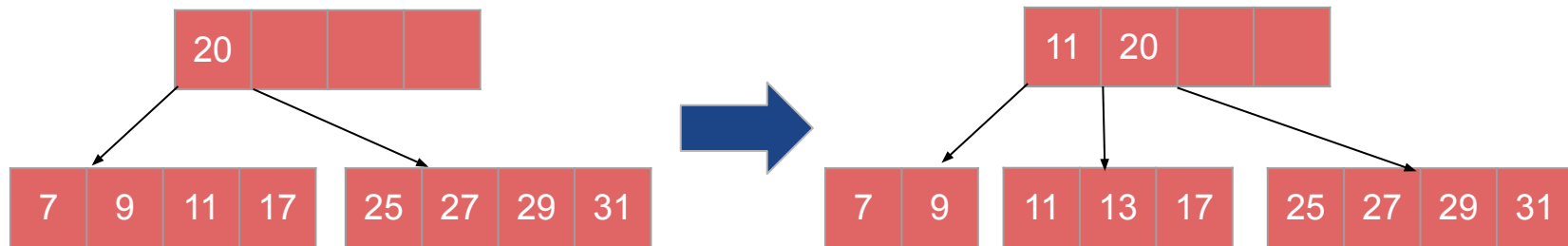
Insertar 13



Inserción

M=5

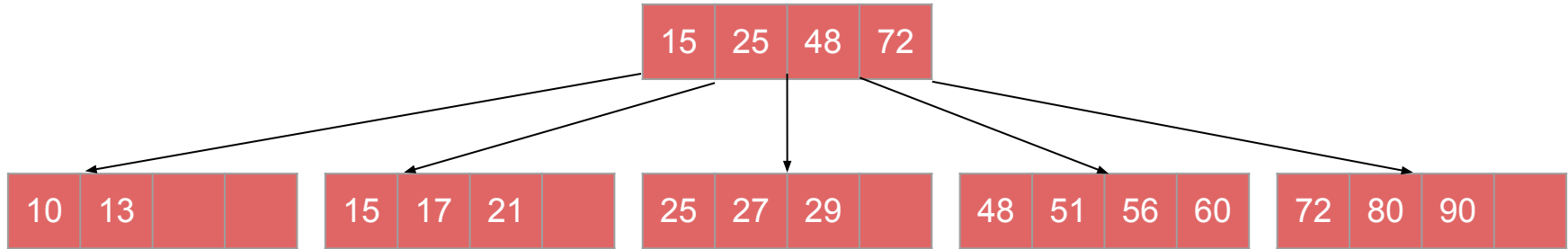
Insertar 13



Inserción - Ejemplo 2

M=5

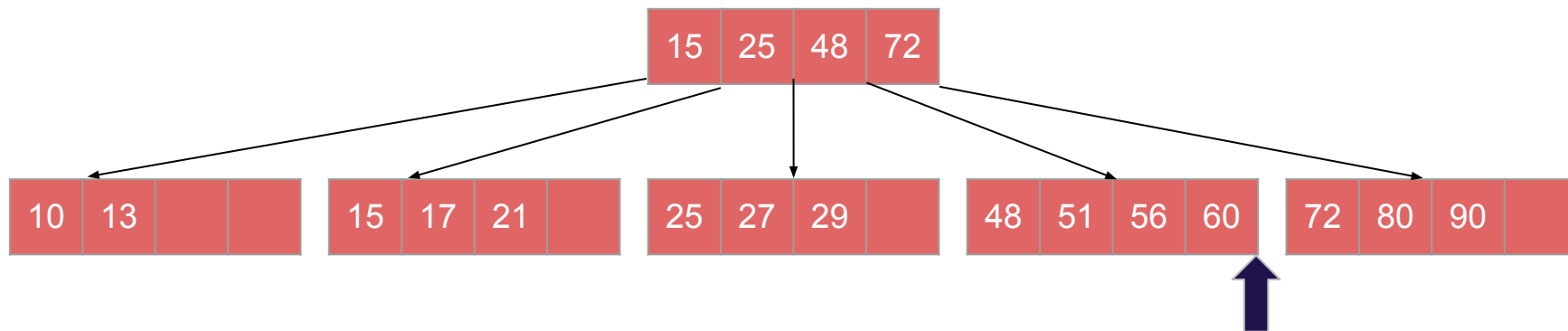
Insertar 67



Inserción - Ejemplo 2

M=5

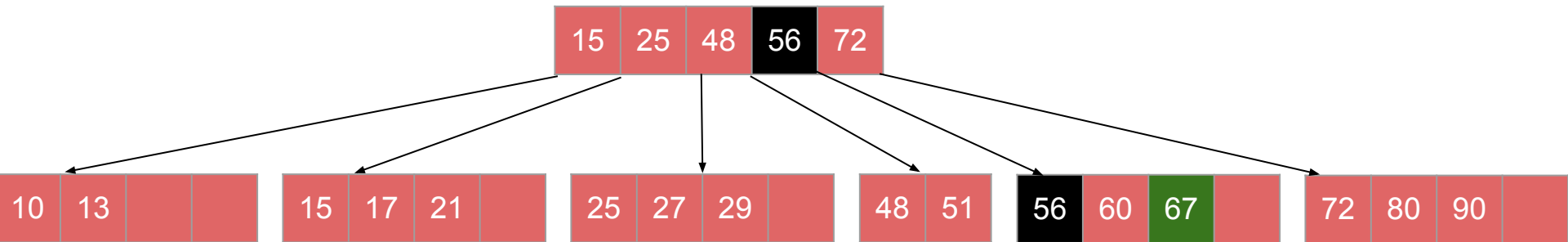
Insertar 67



Inserción - Ejemplo 2

M=5

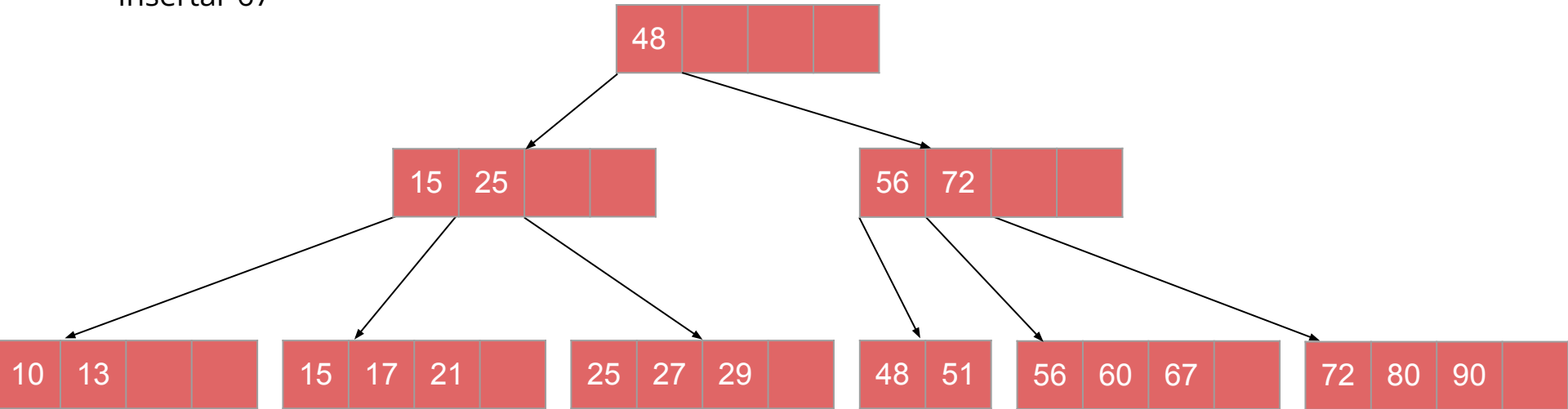
Insertar 67



Inserción - Ejemplo 2

M=5

Insertar 67



Eliminación

La eliminación es un proceso relativamente más sencillo que el de un árbol B.

El dato siempre está en el nodo hoja.

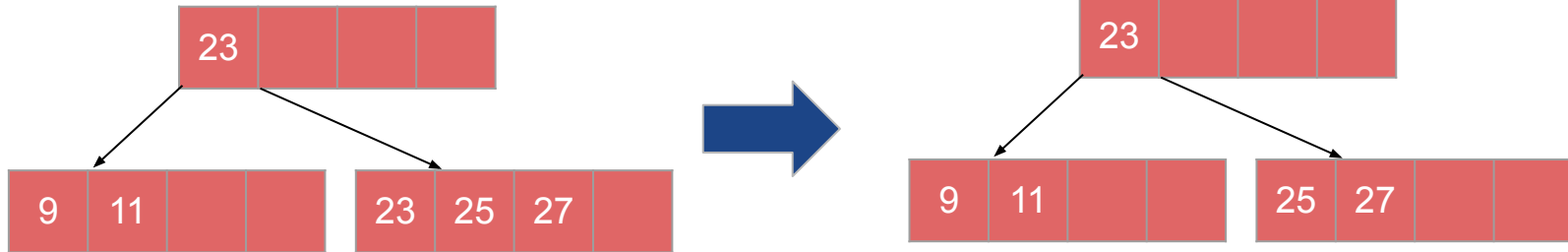
Si al borrar se conserva la regla de $(M/2)-1$ claves, no se debe hacer ningún proceso adicional.

Si al borrar se rompe la regla del mínimo (lower), es necesario realizar una redistribución considerando las hojas y los índices.

Eliminación - Ejemplo 1

M=5

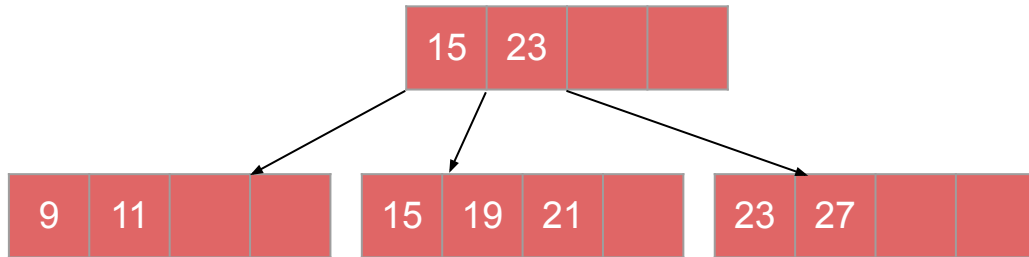
Eliminar 23



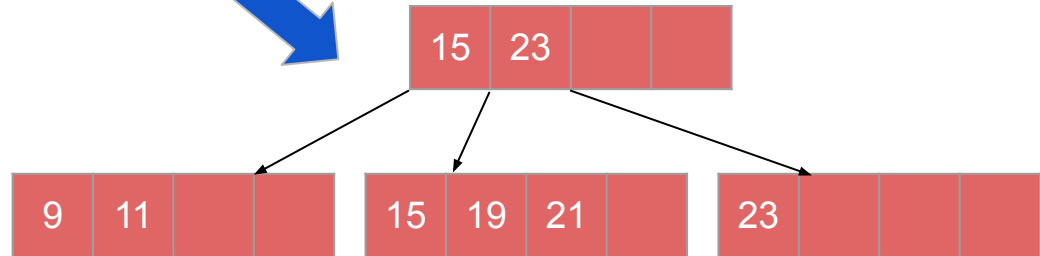
Eliminación - Ejemplo 2

M=5

Eliminar 27



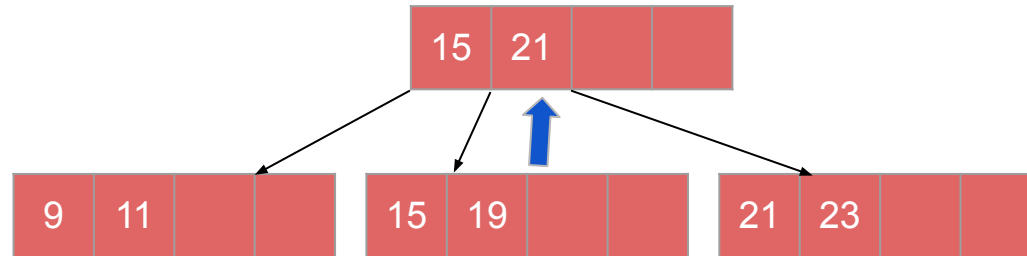
debemos prestar



Eliminación - Ejemplo 2

M=5

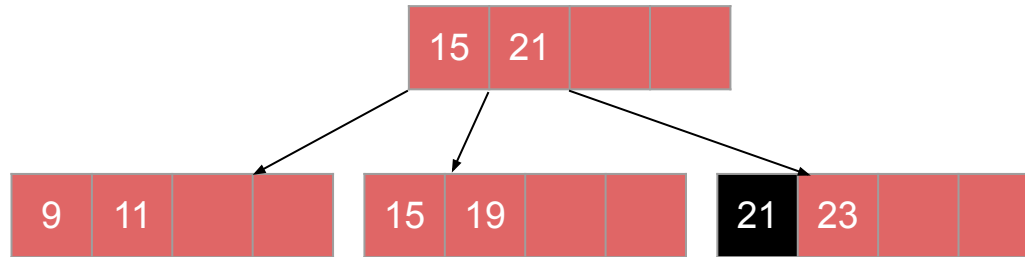
Eliminar 27



Eliminación - Ejemplo 3

M=5

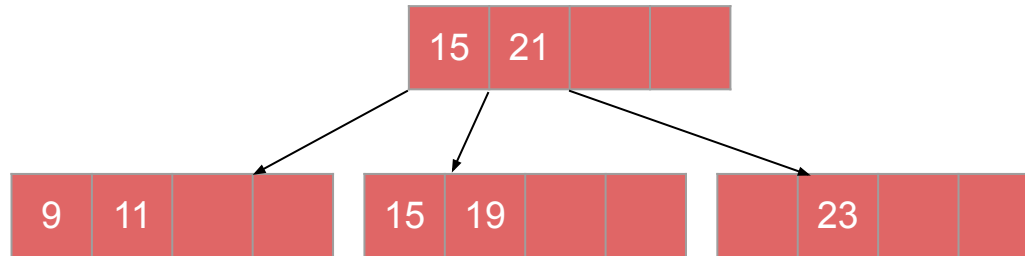
Eliminar 21



Eliminación - Ejemplo 3

M=5

Eliminar 21

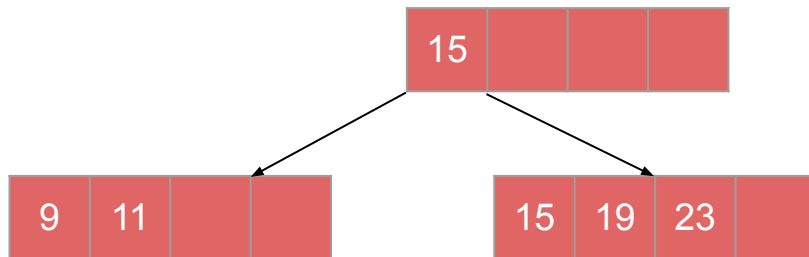


necesitamos fusionar

Eliminación - Ejemplo 3

M=5

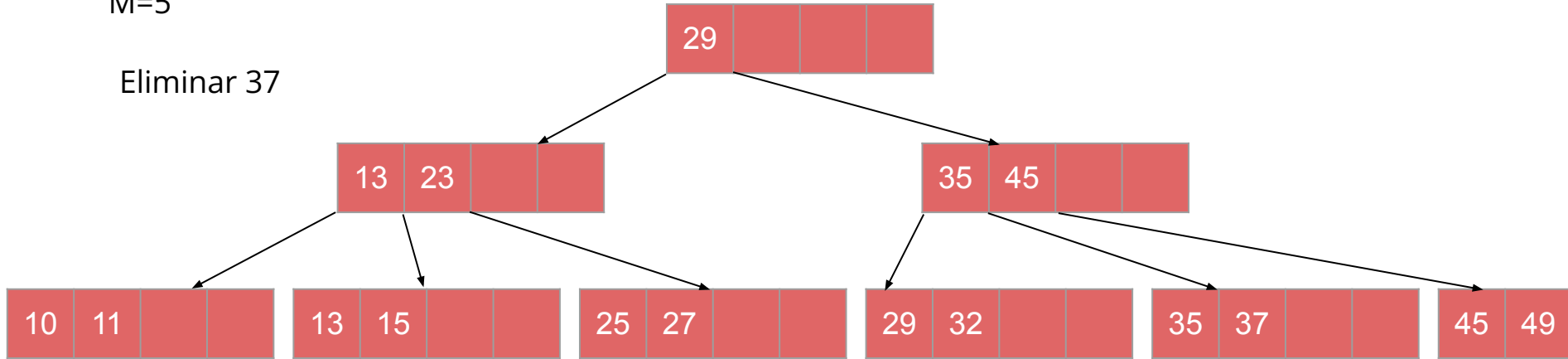
Eliminar 21



Eliminación - Ejemplo 4

M=5

Eliminar 37

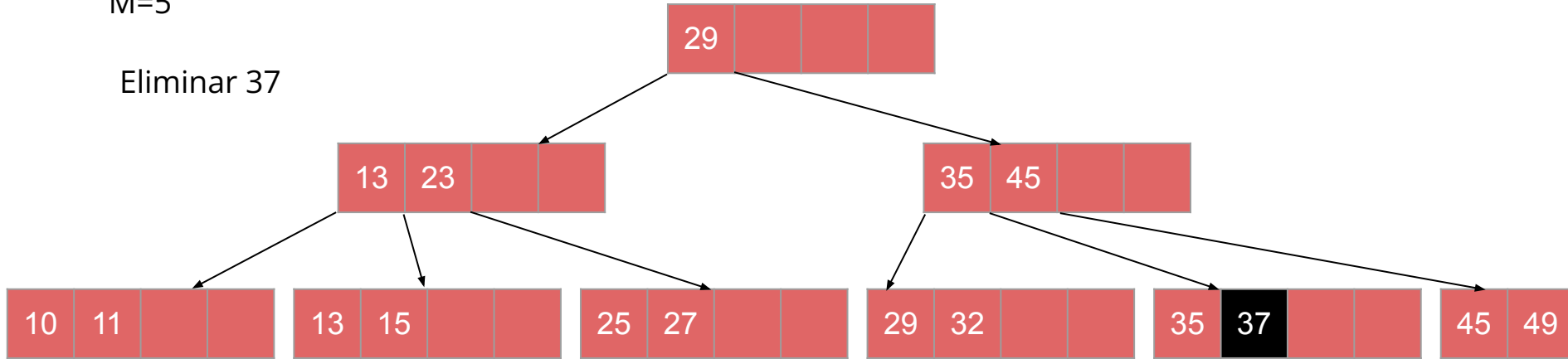


Eliminación - Ejemplo 4

$$l = 2$$
$$ul = 4$$

M=5

Eliminar 37



los vecinos no tienen suficientes
datos para hacer un préstamo
se necesita fusionar

Eliminación - Ejemplo 4

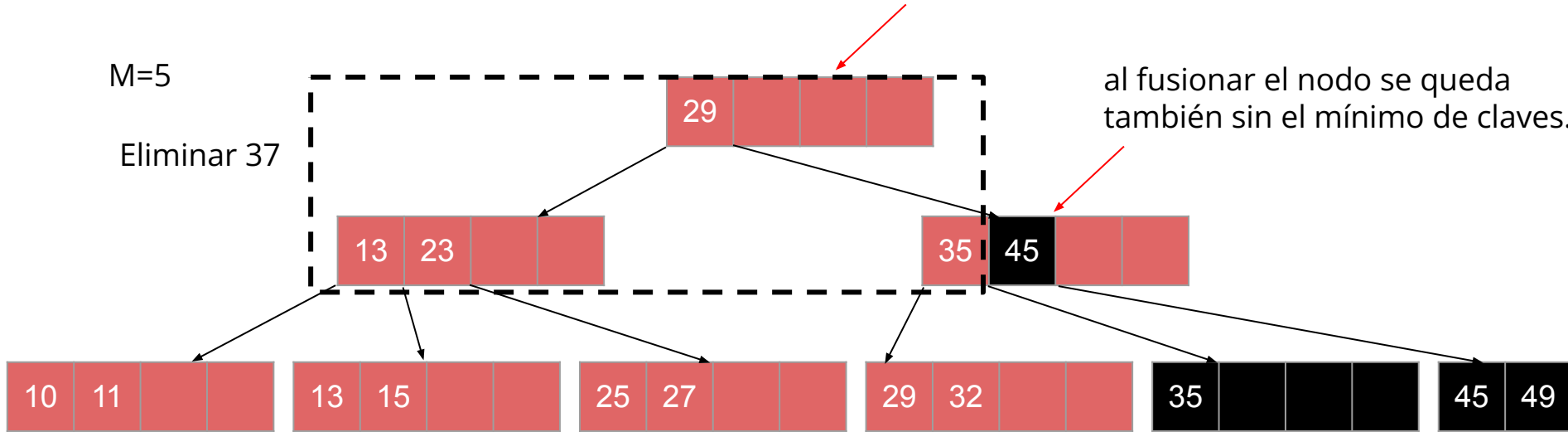
$$l = 2$$
$$u = 4$$

M=5

Eliminar 37

se necesita
fusionar

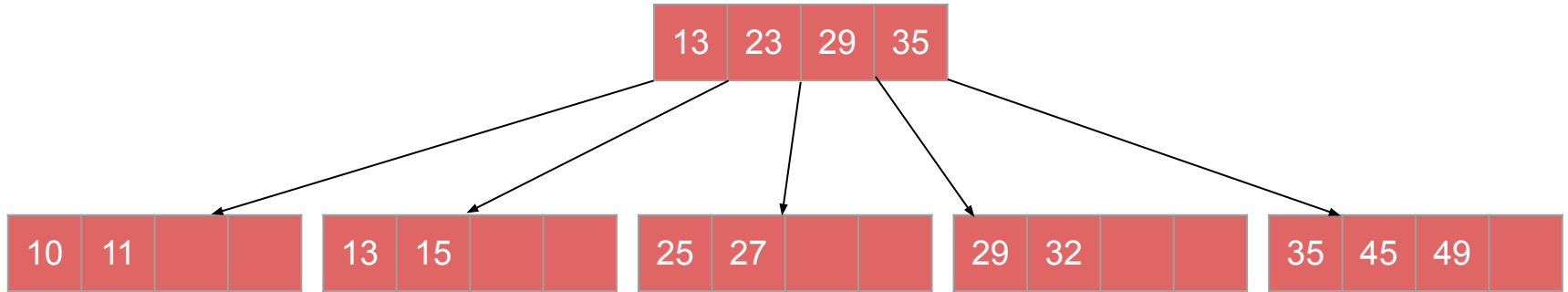
al fusionar el nodo se queda
también sin el mínimo de claves.



Eliminación - Ejemplo 4

M=5

Eliminar 37



Nodos

estructura NodoHoja {

numClaves: Entero;

valores: Objeto[];

claves: Entero[];

clavesUsadas: Entero;

nodolzq : NodoHoja;

nodoDer : NodoHoja;

}

Constructor de un Nodo Hoja (PSEUDO CÓDIGO)

NodoHoja {

constructor NodoHoja(t) {

*valores[(2*t - 1)];*

*claves[(2*t - 1)];*

nodolzq = null;

nodoDer = null;

}

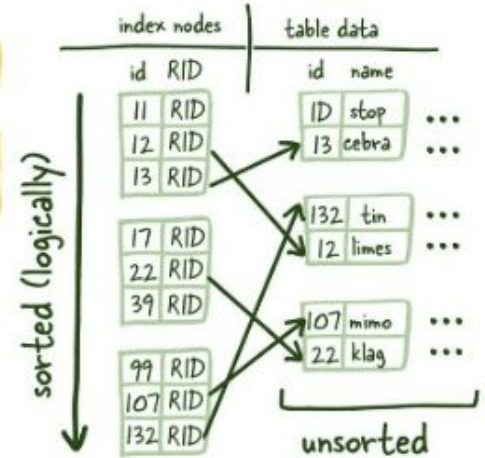
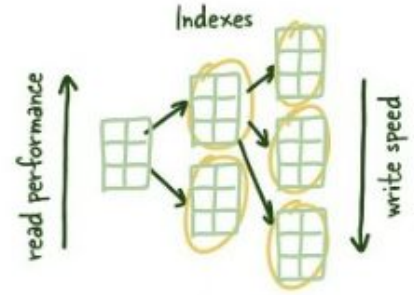
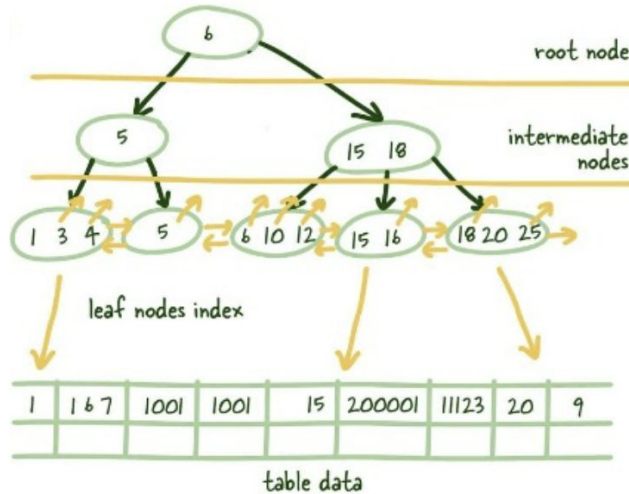
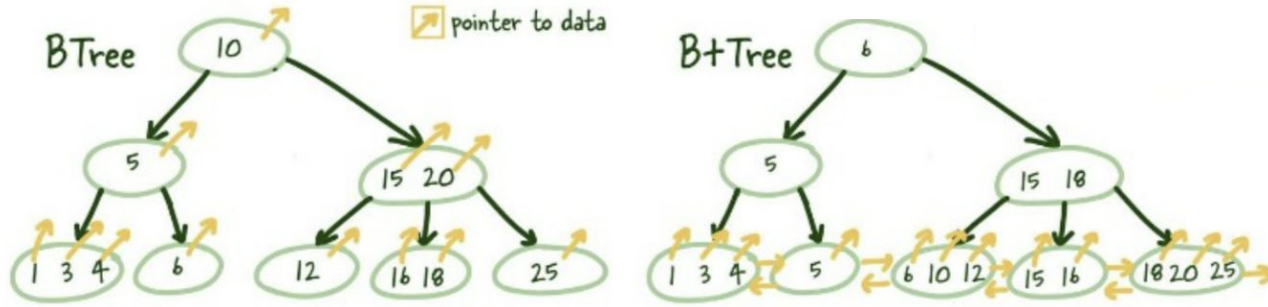
}

Nodos

```
estructura NodoInterno {  
  
    numClaves: Entero;  
  
    claves: Entero[];  
  
    clavesUsadas: Entero;  
  
    nodoPadre : NodoInterno;  
  
    nodosHijo : NodoInterno[];  
  
}
```

```
Constructor de un Nodo Interno NodoInterno {  
  
    constructor NodoInterno(t) {  
  
        clave[(2*t - 1)];  
  
        nodoHijos[(2*t)];  
  
        clavesUsadas=0;  
  
        nodoPadre=null;  
  
    }  
  
}
```

Implementación de Árboles B y B+



Ventajas y Desventajas

Los nodos hoja del árbol están conectados entre sí a través de una lista enlazada.

Esto aumenta el coste de inserción.

Mejora la eficiencia en la búsqueda.

Hoja de Trabajo

Implemente un árbol B+ que realice las siguientes operaciones:

Inserte: 10,17,3,29,4,5,18,6,22,1,33,35

Posteriormente

elimine: 18,33, 29

Grado: 5