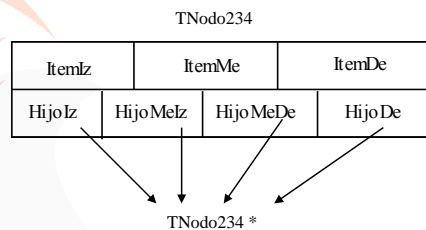


3.4. Arbres 2-3-4

DEFINICIONS

- Un arbre 2-3-4 és un arbre que està buit o satisfà les propietats següents:
 - Els nodes poden tindre 2, 3 o 4 fills (2-node, 3-node o 4-node)
 - Compleix les propietats d'arbre multicamí de busca
 - Totes les fulles estan en el mateix nivell
- Representació



```
class TArb234 {
public:
    ....
private:
    TNodo234 * farb;
};
```

1

3.4. Arbres 2-3-4

OPERACIONS BÀSIQUES. PROPIETATS

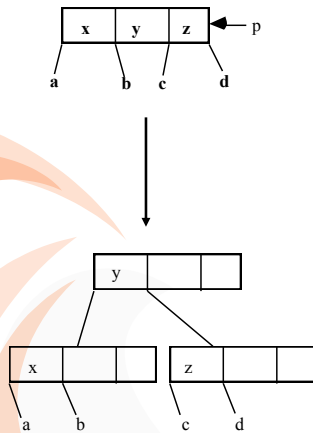
- Operacions bàsiques:
 - Busca (semblant als arbres multicamí de busca)
 - Inserció (es fa en les fulles. Es poden produir reestructuracions de l'arbre)
 - Esborrat (es fa en les fulles. Es poden produir reestructuracions de l'arbre)
- Propietats:
 - En un arbre 2-3-4 d'altura h tenim:
 - $2^h - 1$ elements si tots els nodes són del tipus 2-node
 - $4^h - 1$ elements si tots els nodes són del tipus 4-node
 - de manera que l'altura d'un arbre 2-3-4 amb n elements es troba entre els límits: $\log_4 (n+1)$ i $\log_2 (n+1)$
 - Les reestructuracions es fan des de l'arrel cap a les fulles

2

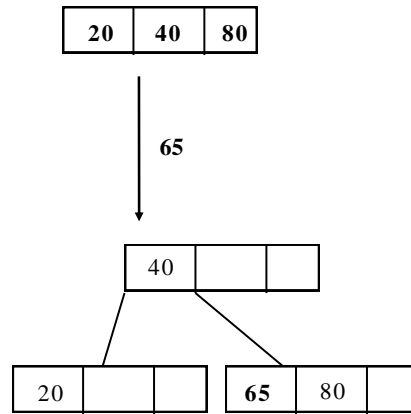
3.4. Arbres 2-3-4

OPERACIONS BÀSIQUES. INSERCIÓ (I)

- Hi ha 3 situacions en què es pot trobar un 4-node: (1)



És l'arrel d'un arbre 2-3-4:
(DIVIDERAIZ (p))

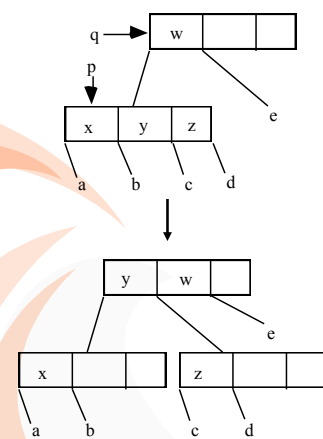


3

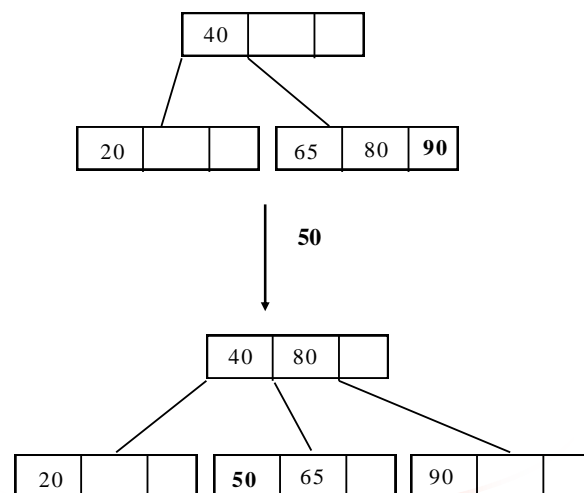
3.4. Arbres 2-3-4

OPERACIONS BÀSIQUES. INSERCIÓ (II)

- Hi ha 3 situacions en què es pot trobar un 4-node: (2)



El pare (q) és un 2-node:
(DIVIDEHIJODE2 (p, q))

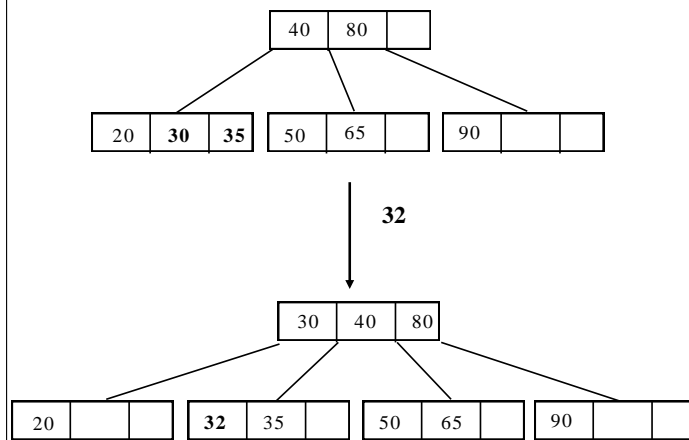
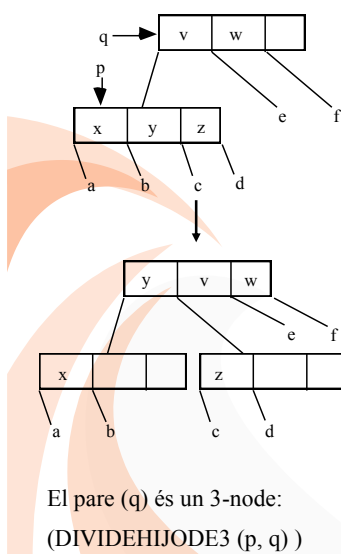


4

3.4. Arbres 2-3-4

OPERACIONS BÀSIQUES. INSERCIÓ (III)

- Hi ha 3 situacions en què es pot trobar un 4-node: (3)



5

3.4. Arbres 2-3-4

OPERACIONS BÀSIQUES. INSERCIÓ (IV)

ALGORITMO insertar (A: TArb234, y: item)

si arrel es 4-nodo → DIVIDERAIZ

si en el camí fins la fulla me trobe un 4-nodo → DIVIDEHIJO

6

3.4. Arbres 2-3-4

OPERACIONS BàSIQUES. INSERCIÓ (V)

```

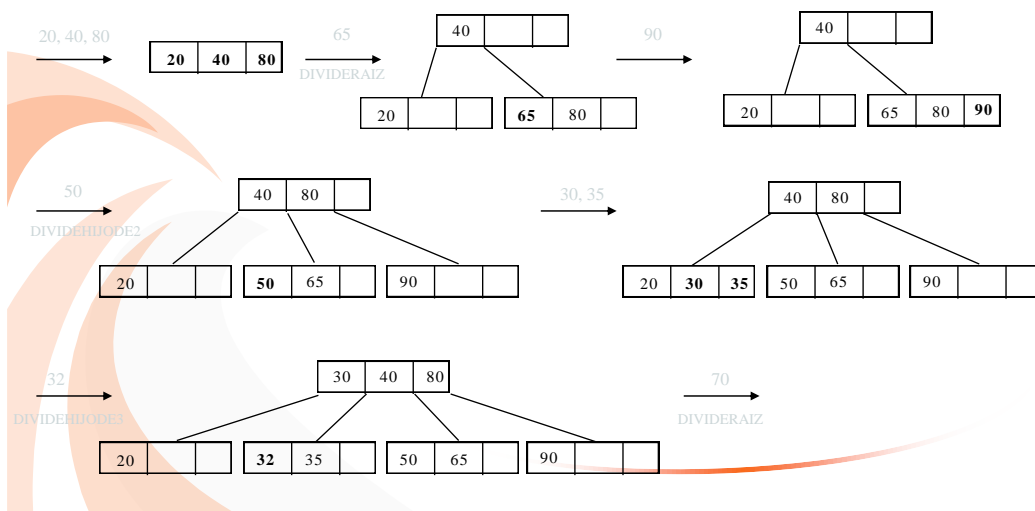
ALGORITMO insertar (A: TArb234, y: item)
VAR p, q: TNode234*; noencontrado: Boolean; B: TArb234; FVAR
  p = A.farb; q = p;
  si EsVacio( A ) entonces A = ENRAIZAR(A, y, B)
  sino
    si p es 4-nodo entonces DIVIDERAIZ( A ) fsi
    noencontrado = VERDADERO;
    mientras noencontrado hacer
      si p es 4-nodo entonces
        si q es 2-nodo entonces DIVIDEHIJODE2( p, q );
        sino DIVIDEHIJODE3( p, q ); fsi
        p = q;
      fsi
    caso de COMPARAR( y, p ):
      0:// Clave de y coincide con clave en p
        ERROR, ETIQUETA EXISTENTE;
      1:// p apunta a un nodo hoja
        PONER( y, p ); noencontrado = FALSO;
      2:// clave( y ) < ItemIz.clave( p )
        q = p; p = p → HiIz;
      3:// ItemIz.clave(p)<clave(y)<ItemMe.clave(p)
        q = p; p = p → HiMeIz;
      4://ItemMe.clave(p)<clave(y)<ItemDe.clave(p)
        q = p; p = p → HiMeDe;
      5:// clave(y) > ItemDe.clave(p)
        q = p; p = p → HiDe;
    fcaso
      fmientras
    fsi
  FALGORITMO
  
```

7

3.4. Arbres 2-3-4

OPERACIONS BàSIQUES. INSERCIÓ. EXEMPLE (VI)

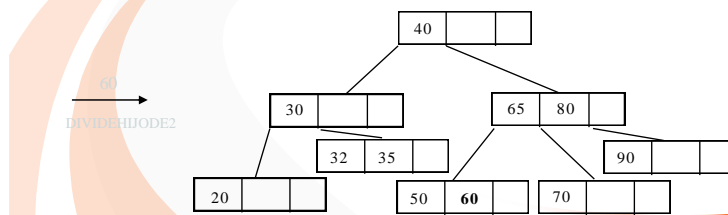
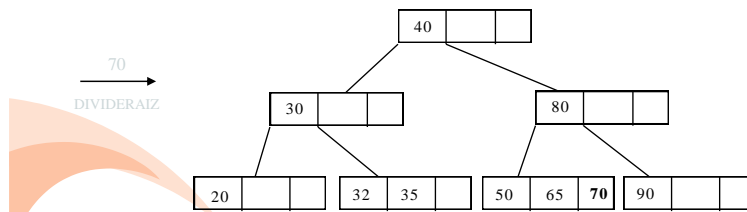
✱ **Exemple.** Insertar en un arbre 2-3-4 buit els següents ítems: 20, 40, 80, 65, 90, 50, 30, 35, 32, 70, 60



8

3.4. Arbres 2-3-4

OPERACIONS BÀSIQUES. INSERCIÓ. EXEMPLE (IV)

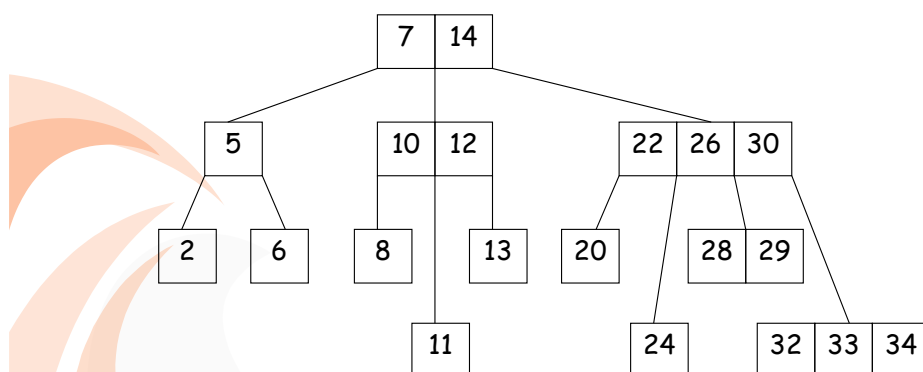


9

3.4. Arbres 2-3-4

EXERCICIS *inserció*

- 1) Donat el següent arbre 2-3-4, inserir els elements 21 i 35



10

3.4. Arbres 2-3-4

OPERACIONS BÀSIQUES. ESBORRAT (I)

- Es redueix a l'esborrat d'un element en una fulla
- En el moviment de busca, quan passem a un node en el següent nivell, aquest node ha de ser 3-node o 4-node; si no és així (és 2-node), cal reestructurar
 - p = node on estem
 - q = següent node en la busca
 - r = un dels nodes adjacents a q (si n'hi ha dos d'adjacents, triem r segons criteri –germà de l'esquerra o germà de la dreta–)

12

3.4. Arbres 2-3-4

OPERACIONS BÀSIQUES. ESBORRAT (II)

- Algoritme d'Esborrat d'un element
 - Començar p = arrel
 - Escollir q = següent en el camí de búsqueda
 - Reestructuracions si és necessari // q és 2-node
 - Mentre q no siga un node fulla o q siga 2-node
 - $p = q$
 - q = següent en el camí de búsqueda
 - Reestructuracions si és necessari // q és 2-node
 - Borrar el elemento // q està en un node fulla i q no és 2-node
- Casos:
 1. p és una fulla: p només pot ser 2-node si és l'arrel
 2. q és 3-node o 4-node: la búsqueda continua en q sense reestructurar

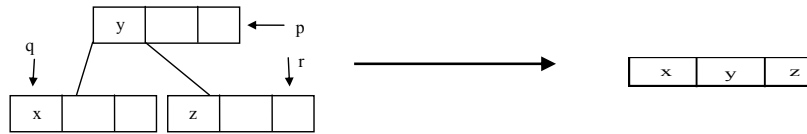
13

3.4. Arbres 2-3-4

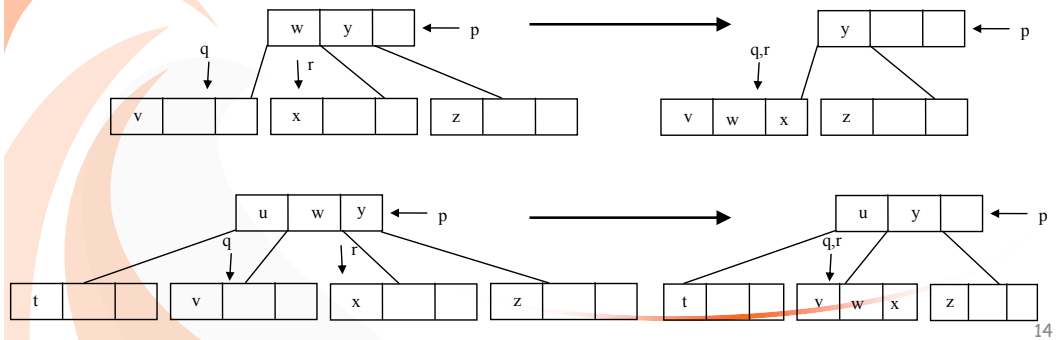
OPERACIONS BÀSIQUES. ESBORRAT (III)

3. q és 2-node i r és 2-node (COMBINACIÓ):

1. p és 2-node: és l'arrel



2. p és 3-node o 4-node

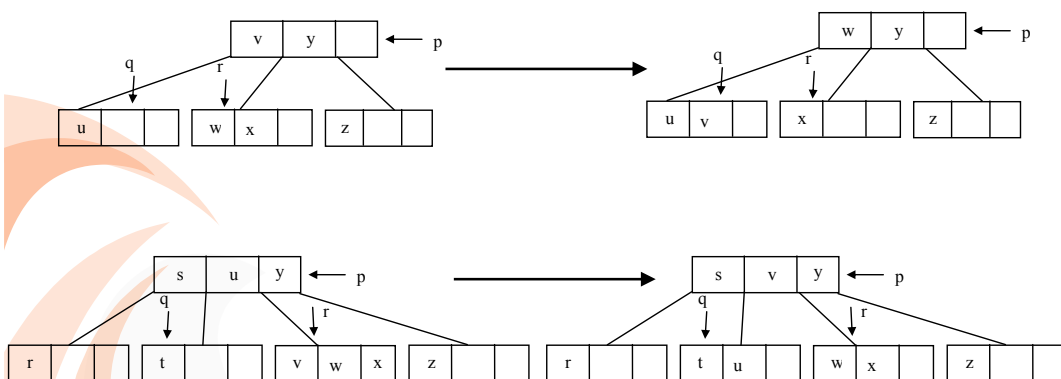


14

3.4. Arbres 2-3-4

OPERACIONS BÀSIQUES. ESBORRAT (IV)

4. q és 2-node i r és 3-node o 4-node (ROTACIÓ):

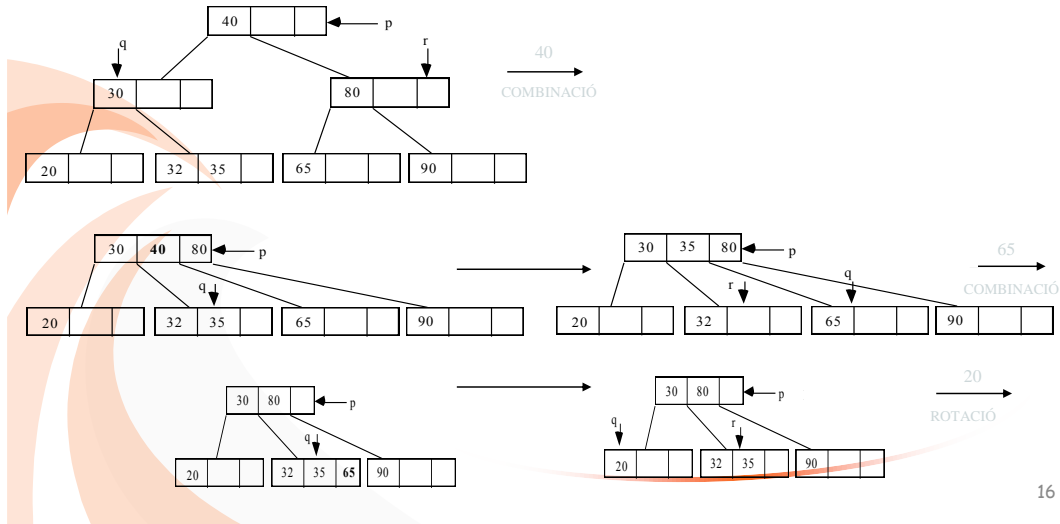


15

3.4. Arbres 2-3-4

OPERACIONS BÀSIQUES. ESBORRAT. EXEMPLE (V)

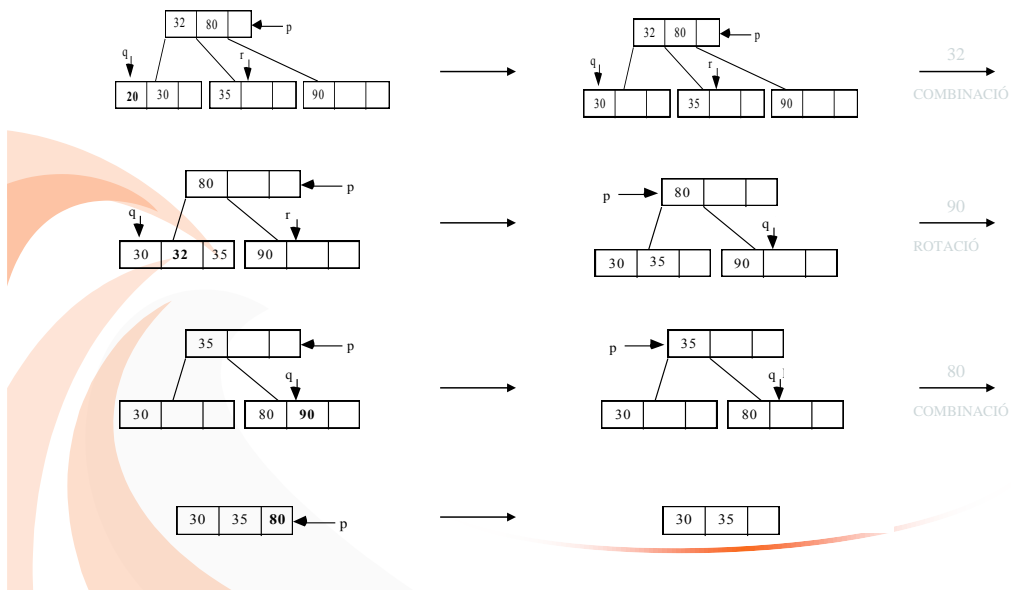
✳ **Exemple.** Esborrar en el següent arbre 2-3-4 els següents ítems: 40, 65, 20, 32, 90, 80. (Criteris: (1) si el node té dos fills, cal substituir pel major de l'esquerra; (2) Si hi ha dos nodes adjacents a q, llavors r serà el germà de l'esquerra)



16

3.4. Arbres 2-3-4

OPERACIONS BÀSIQUES. ESBORRAT. EXEMPLE (V)

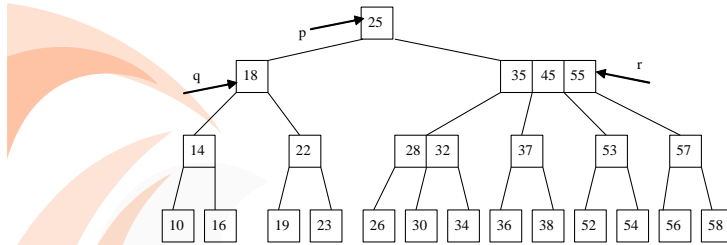


17

3.4. Arbres 2-3-4

EXERCICIS esborrat

- 1) Esborreu en el següent arbre 2-3-4 l'item: 25, 18, 55 y 35. (Criteris: (1) si el node té dos fills, cal substituir pel major de l'esquerra; (2) si hi ha dos nodes adjacents a q, llavors r serà el germà de l'esquerra)



18

3.4. Arbres 2-3-4

Preguntes de tipus test: Vertader vs. Fals

- L'arbre 2-3-4 no buit té com a mínim una clau en cada node
- La complexitat temporal en el pitjor cas de l'operació inserció en un arbre 2-3-4 és $\log_2(n+1)$
- Un arbre 2-3-4 és un arbre binari complet

23

Arbres de búsqueda

Aplicacions

- **Accés a grans fitxers de dades. Organització interna d'una BD.**
 - Restricció: Els índexos resideixen en disc.
 - Problema: Accessos a disc molt costosos.
 - Solució: Organitzar arbres amb múltiples claus “n” per node
 - Recuperar un node d'aquest arbre (un accés al fitxer índex) selecciona una entre “n” alternatives, front a una entre dues (Arbre Binari).
 - L'índex pot dissenyar-se perquè la grandària de cada node coincidisca amb el d'un bloc de disc.