

Árboles

EEDD - GRADO EN ING. INFORMÁTICA - UCO

Árboles B

Contenidos

- Concepto de Árbol B.
- Operaciones de Inserción y borrado.

EEDD - GRANDO EN ING. INFORMÁTICA - UCO

Árboles B

- Motivación.
 - Los árboles multcamino reducen la altura del árbol aumentando el su grado.
 - Necesitamos estrategias para asegurar que el árbol se mantiene equilibrado tras las inserciones y eliminaciones de claves.

Árboles B

- Definición.
 - Es un árbol de búsqueda multicamino de grado D que mantiene además el invariante:
“Todo nodo tiene al menos $\lfloor (D-1)/2 \rfloor$ claves (excepto la raíz que puede tener menos).”
 - Todos los nodos hoja están en el mismo nivel.
- $\{5, 13, 20, 16, 27, 12, 23, 25, 29, 32\}$ y grado 5.

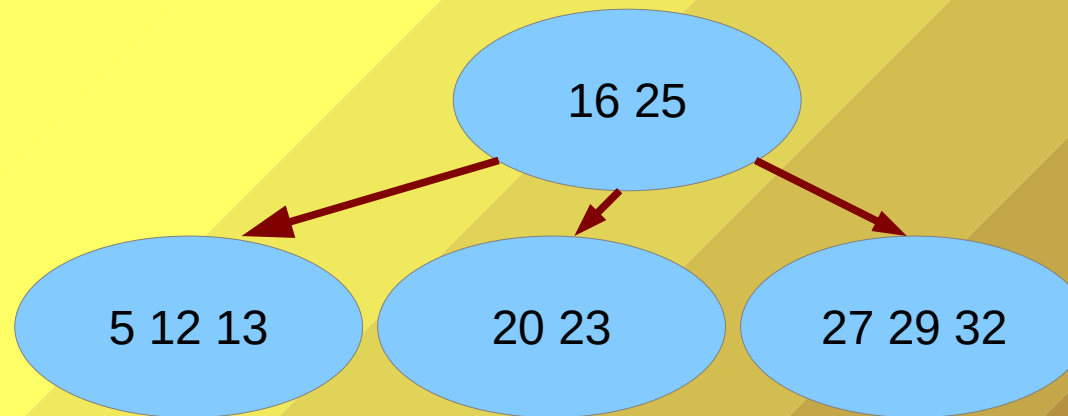


Árboles B

- Definición.
 - Es un árbol de búsqueda multicamino de grado D que mantiene además el invariante:

“Todo nodo tiene al menos $\lfloor (D-1)/2 \rfloor$ claves (excepto la raíz que puede tener menos).”

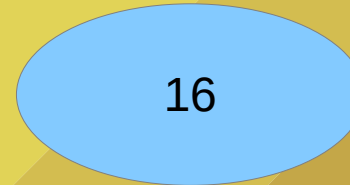
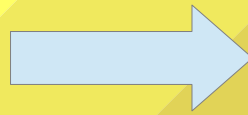
- Todos los nodos hoja están en el mismo nivel.
- {5,13,20,16,27,12,23,25,29,32} y grado 5.



Árboles B

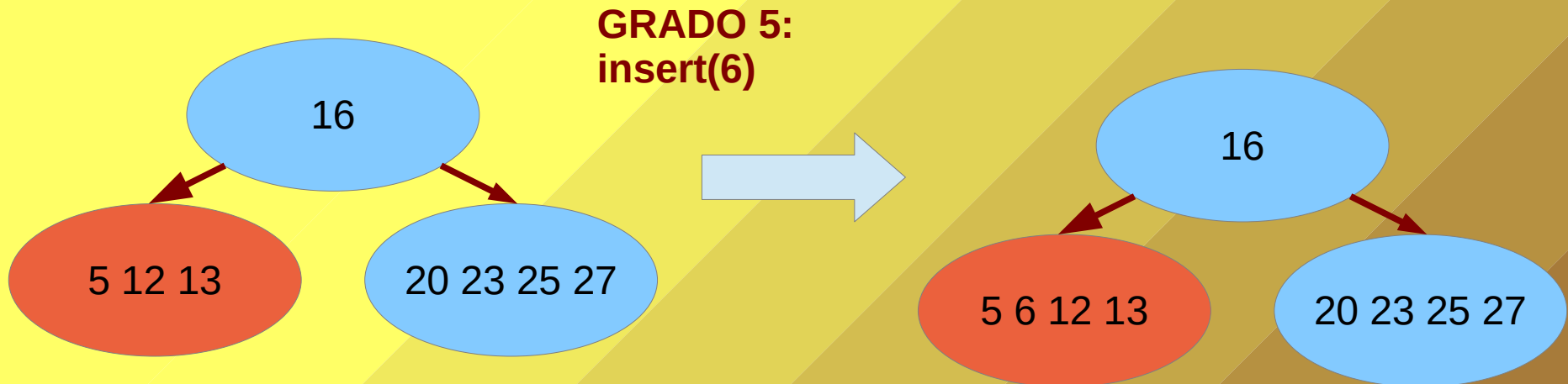
- Inserción en árbol B: caso 0.
 - El árbol está vacío. Se crea el nodo raíz con la clave.
 - Tiene menos de $\lfloor (D-1)/2 \rfloor$ claves, pero es el nodo raíz.

GRADO 5:
insert(16)



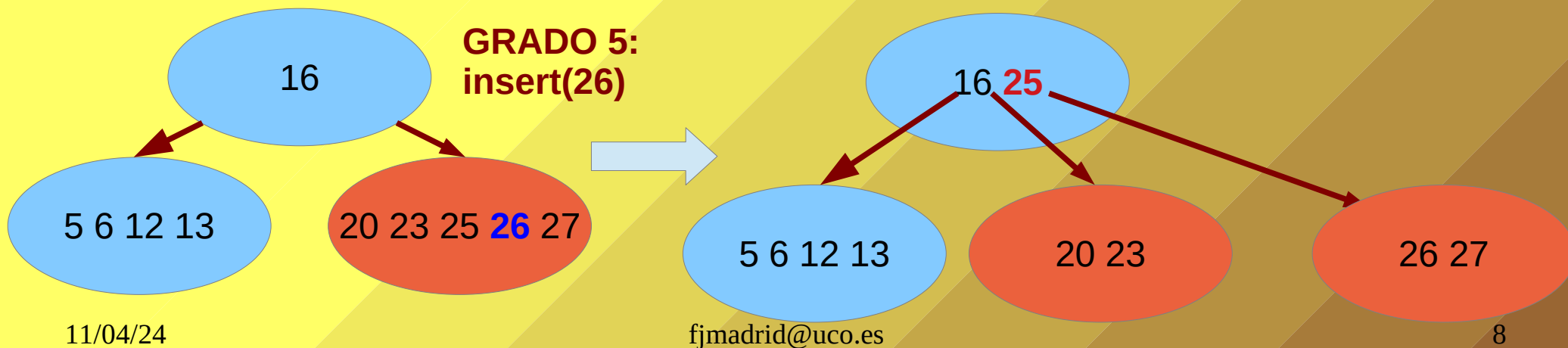
Árboles B

- Inserción en árbol B: caso 1.
 - Aplicar algoritmo de búsqueda MTree para localizar el nodo hoja que le corresponde.
 - Este nodo no está lleno. Se insertar en orden.



Árboles B

- Inserción en árbol B: caso 2 (overflow) (paso 1).
 - Al insertar, la hoja se satura: (tendremos D claves).
 - Dividirlo en tres partes:
 - mediana -> insertar en el nodo padre (si no tiene, primero crear un nodo nuevo que será la nueva raíz).
 - $\lfloor (D-1)/2 \rfloor$ claves menores -> nueva hoja izquierda de la mediana.
 - $\lfloor (D-1)/2 \rfloor$ claves mayores -> nueva hoja derecha de la mediana.

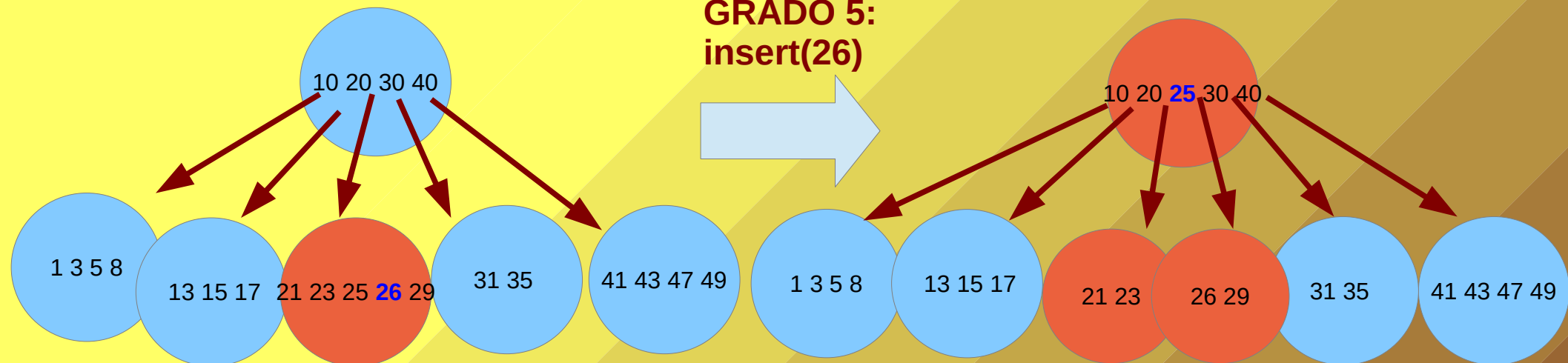
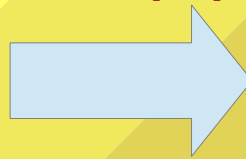


Árboles B

- Inserción en árbol B: caso 2 (overflow) (paso 2).
 - Comprobar en el padre si este se ha saturado también tras insertar la mediana.
 - Repetir el proceso de división e inserción de la mediana en el padre (si es necesario) en el camino hasta la raíz.

EEDD - GRANDO EN ING. INFORMÁTICA - UCO

GRADO 5:
insert(26)



Árboles B

- Inserción en árbol B: caso 2 (overflow) (paso 2).
 - Comprobar en el padre si este se ha saturado también tras insertar la mediana.
 - Repetir el proceso de división e inserción de la mediana en el padre (si es necesario) en el camino hasta la raíz



Árboles B

- Inserción: {5,13,20,16,27,12,23,25,29,32}

EEDD - GRANDO EN ING. INFORMÁTICA - UCO

GRADO 5

Árboles B

- Inserción: {5,13,20,16,27,12,23,25,29,32}

1: Insert 5, 13, 20, 16

5 13 16 20

EEDD - GRADO EN ING. INFORMÁTICA - UCO

GRADO 5

Árboles B

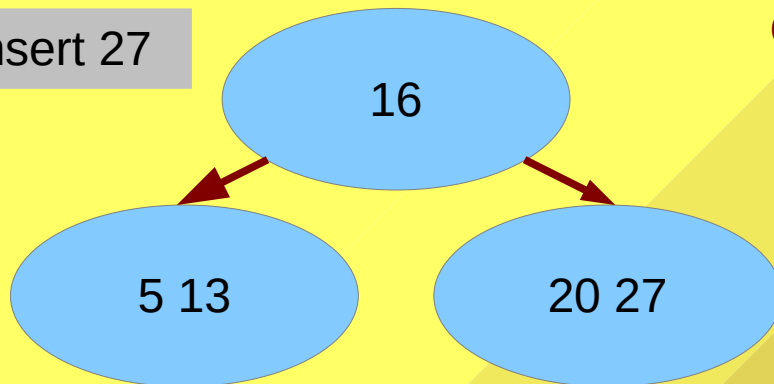
- Inserción: {5,13,20,16,27,12,23,25,29,32}

1: Insert 5, 13, 20, 16

5 13 16 20

2: Insert 27

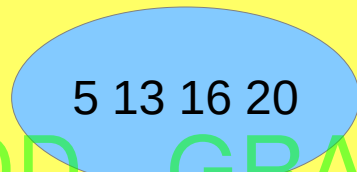
GRADO 5



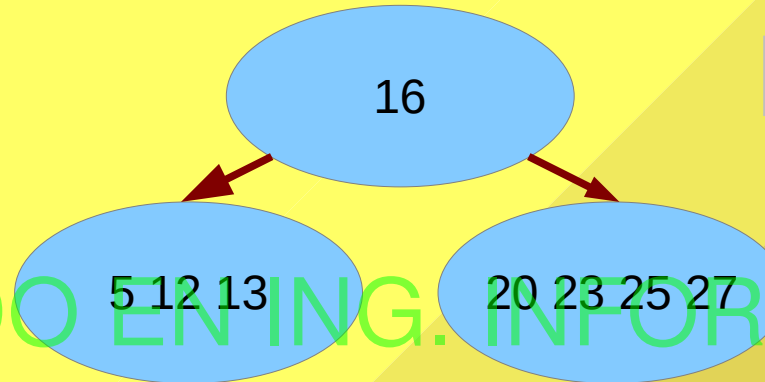
Árboles B

- Inserción: {5,13,20,16,27,12,23,25,29,32}

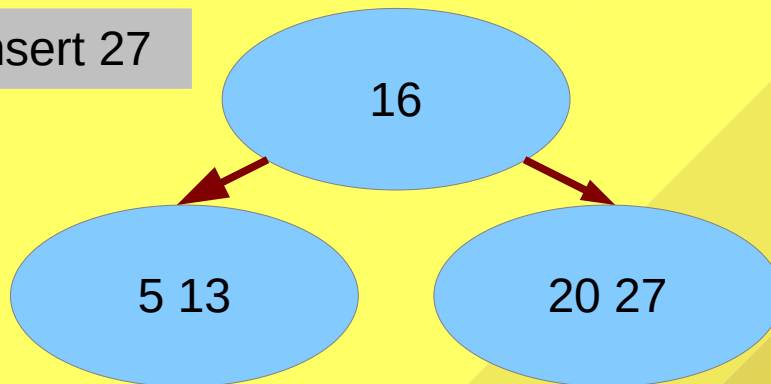
1: Insert 5, 13, 20, 16



3: Insert 12, 23, 25



2: Insert 27

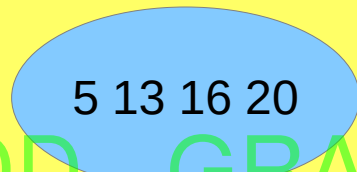


GRADO 5

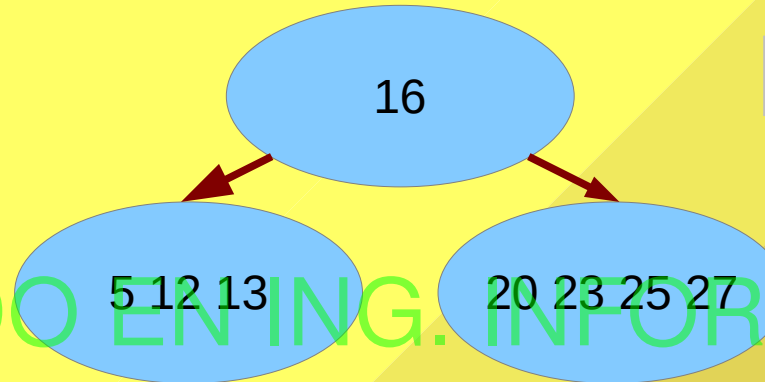
Árboles B

- Inserción: {5,13,20,16,27,12,23,25,29,32}

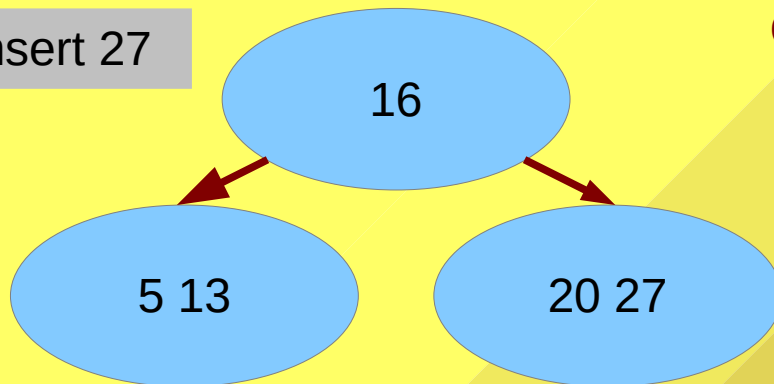
1: Insert 5, 13, 20, 16



3: Insert 12, 23, 25

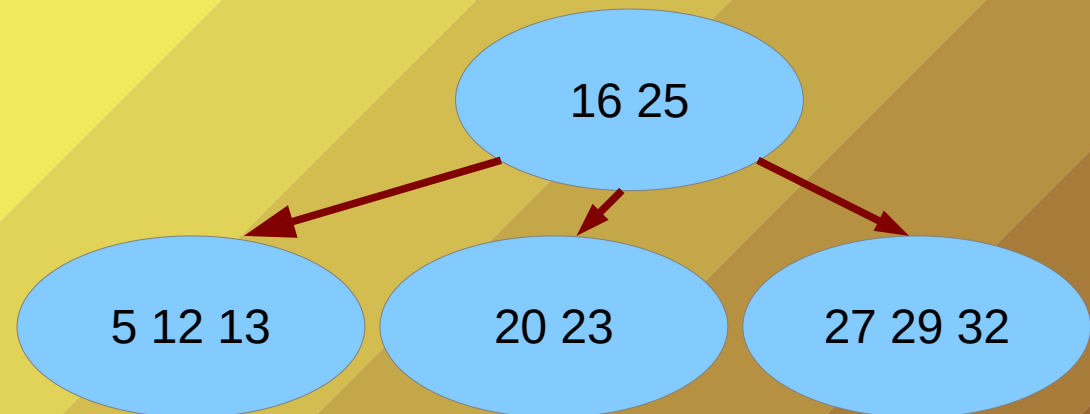


2: Insert 27



GRADO 5

4: Insert 29, 32

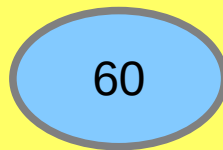


Árboles B

- Borrado en árbol B: Caso 0.
 - Sólo hay un nodo hoja (raíz) y una clave.
 - **Solución:** borrar el nodo y árbol queda vacío.

EEDD - GRADO EN ING. INFORMÁTICA - UCO

Grado 5



remove 60

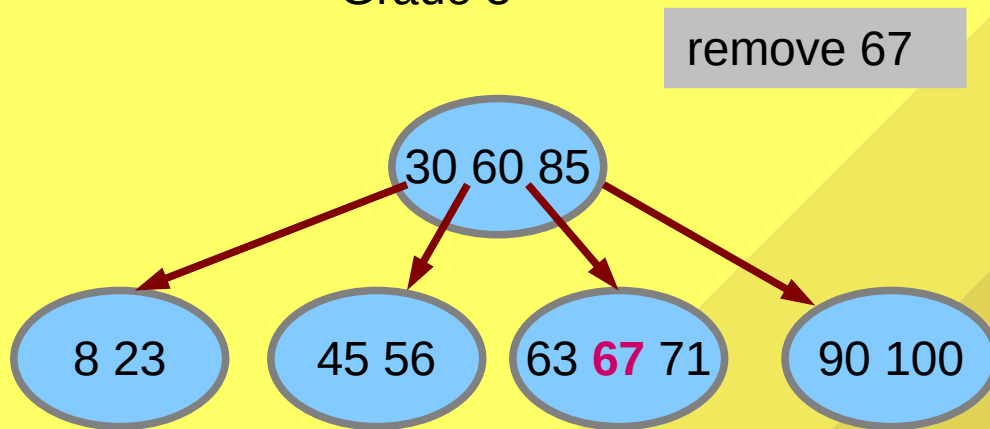


//

Árboles B

- Borrado en árbol B: Caso 1.
 - El cursor está en un nodo hoja con más de $\lfloor (D-1)/2 \rfloor$ claves.
 - **Solución:** borrar la clave.

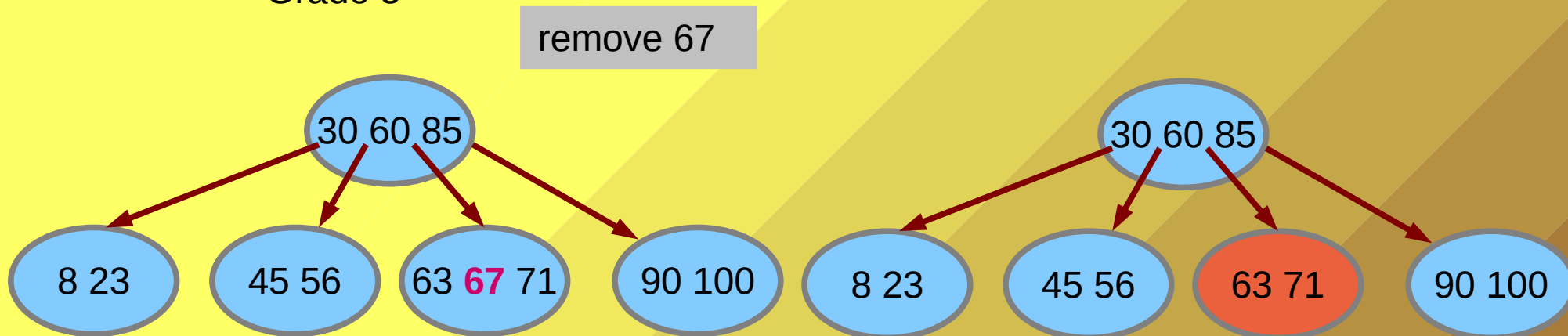
EEDD - GRANDO EN ING. INFORMÁTICA - UCO



Árboles B

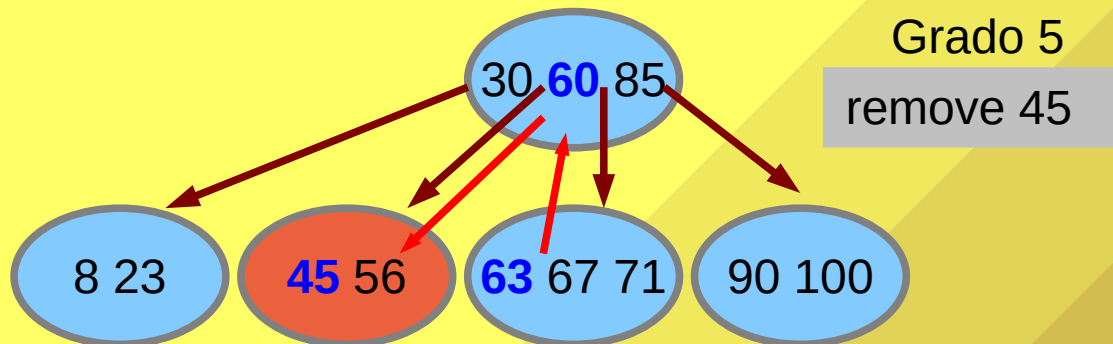
- Borrado en árbol B: Caso 1.
 - El cursor está en un nodo hoja con más de $\lfloor (D-1)/2 \rfloor$ claves.
 - **Solución:** borrar la clave.

EEDD - GRANDO EN ING. INFORMÁTICA - UCO



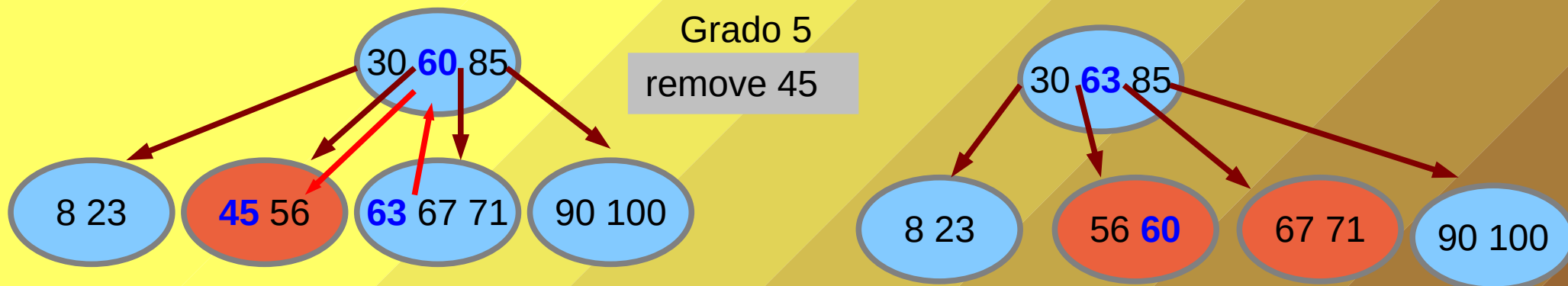
Árboles B

- Borrado en árbol B: Caso 2 (underflow).
 - En cursor está en un nodo hoja con $\lfloor (D-1)/2 \rfloor$ claves.
 - Hay un hermano mayor (menor) con más de $\lfloor (D-1)/2 \rfloor$.
 - **Solución:** El hermano mayor (menor) aporta la menor (mayor) clave que subirá al nodo padre como clave separadora y la anterior clave separadora del padre se insertará en orden en el nodo hoja para impedir quede en situación de underflow.



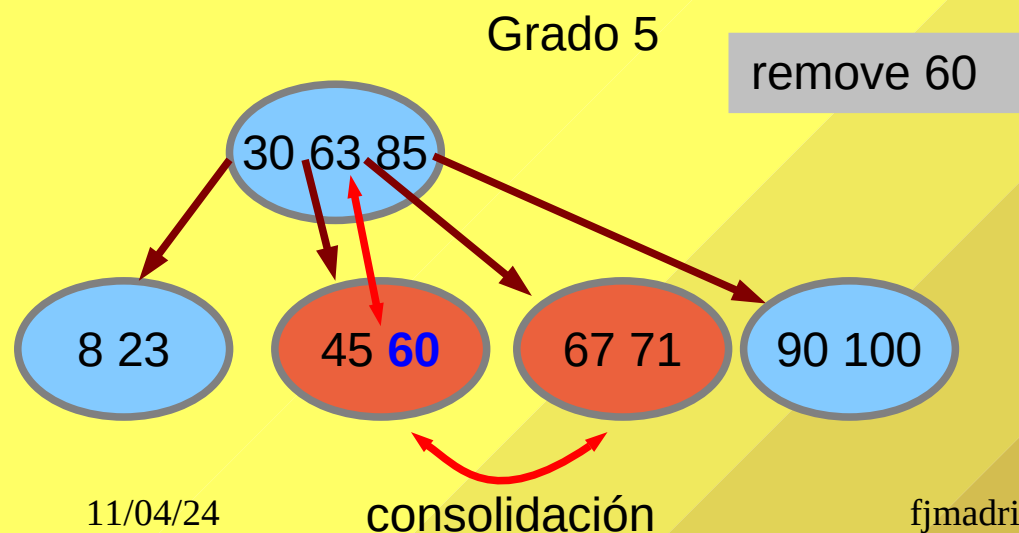
Árboles B

- Borrado en árbol B: Caso 2 (underflow).
 - En cursor está en un nodo hoja con $\lfloor (D-1)/2 \rfloor$ claves.
 - Hay un hermano mayor (menor) con más de $\lfloor (D-1)/2 \rfloor$.
 - **Solución:** El hermano mayor (menor) aporta la menor (mayor) clave que subirá al nodo padre como clave separadora y la anterior clave separadora del padre se insertará en orden en el nodo hoja para impedir quede en situación de underflow.



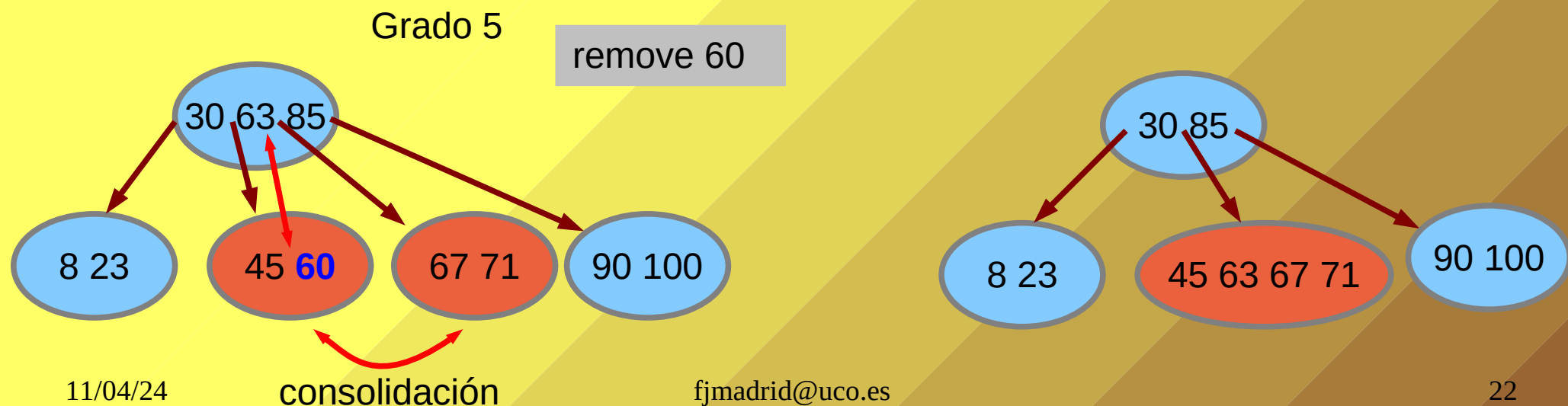
Árboles B

- Borrado en árbol B: Caso 3 (consolidación).
 - Caso 2 pero no hay hermano con más de $\lfloor (D-1)/2 \rfloor$.
 - **Solución:** coger clave separadora del padre y fusionar con hermano mayor (menor).
 - Si el padre en underflow repetir (iterativamente hasta la raíz) la gestión del estado de underflow.



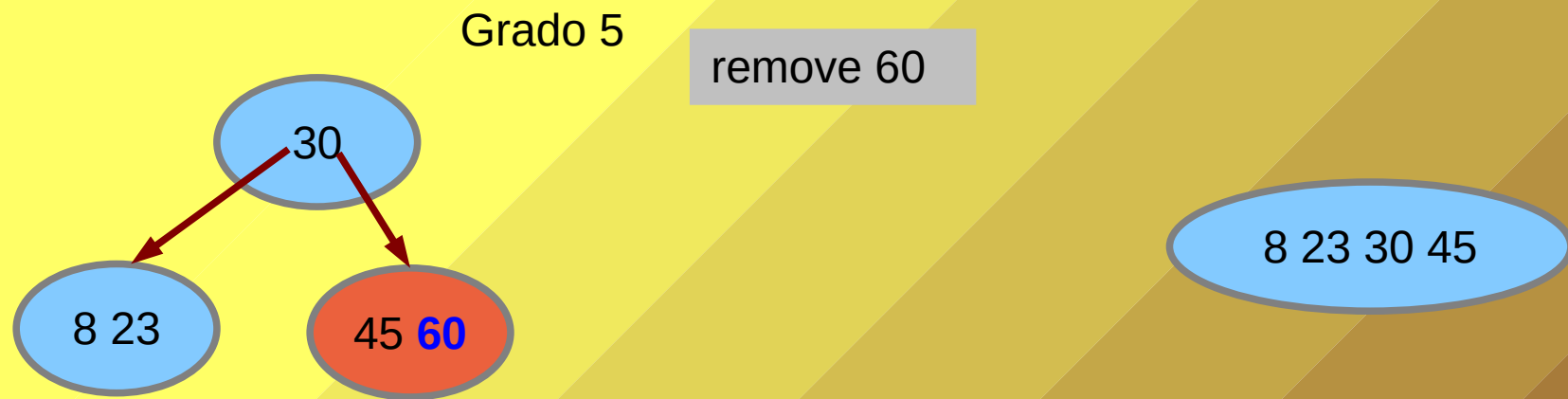
Árboles B

- Borrado en árbol B: Caso 3 (consolidación).
 - Caso 2 pero no hay hermano con más de $\lfloor (D-1)/2 \rfloor$.
 - **Solución:** coger clave separadora del padre y fusionar con hermano mayor (menor).
 - Si el padre en underflow repetir (iterativamente hasta la raíz) la gestión del estado de underflow.



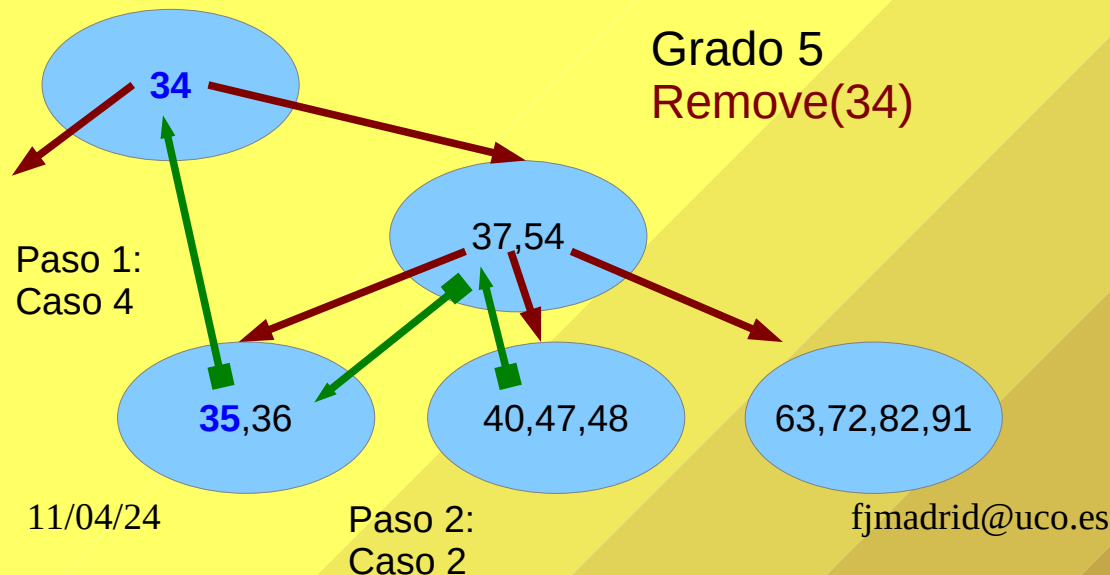
Árboles B

- Borrado en árbol B: Caso 3 (consolidación).
 - Situación especial es cuando el padre es el nodo raíz y solo tiene una clave (recuerda la raíz no tiene porque tener al menos $\lfloor (n-1)/2 \rfloor$ claves).
 - En este caso el árbol reducirá un nivel de altura.



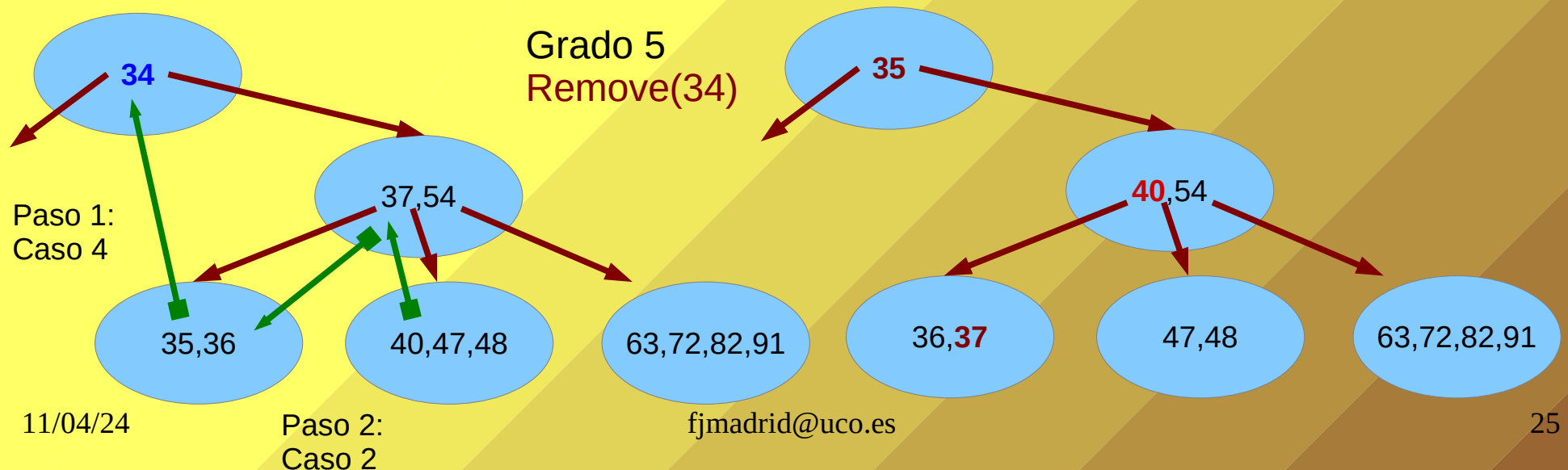
Árboles B

- Borrado en árbol B: Caso 4.
 - El cursor está en un nodo interno.
 - **Solución:** Intercambiar con clave **sucesora** en orden que estará en una hoja. Borrar clave intercambiada en la hoja (casos 1, 2 o 3).



Árboles B

- Borrado en árbol B: Caso 4.
 - Tras aplicar el alg. de búsqueda, el cursor está en un nodo interno.
 - **Solución:** Intercambiar con clave sucesora (predecesora) en orden que estará en una hoja. Borrar clave intercambiada en la hoja (casos 1, 2 o 3).



Árboles B

- Resumiendo.
 - **Ventajas:**
 - El árbol crece en anchura al insertar.
 - Sólo crece en altura cuando se afecta la raíz.
 - El árbol crece de forma equilibrada a costa de que los nodos no estén llenos.
 - **Inconvenientes:**
 - Puede haber muchos nodos no llenos.
 - El procesamiento **secuencial en orden** de clave requerirá de muchos accesos al disco.

Referencias

- Lecturas recomendadas:
 - Cap. 13 de “Estructuras de Datos”, A. Carmona y otros. U. de Córdoba. 1999.
 - Wikipedia:
 - <https://en.wikipedia.org/wiki/B-tree>