Universidad de los Andes

NIVEL 16: ESTRUCTURAS RECURSIVAS N-ARIAS

Árboles n-arios, 1-2-3, 2-3 y B

Universidad de los Andes

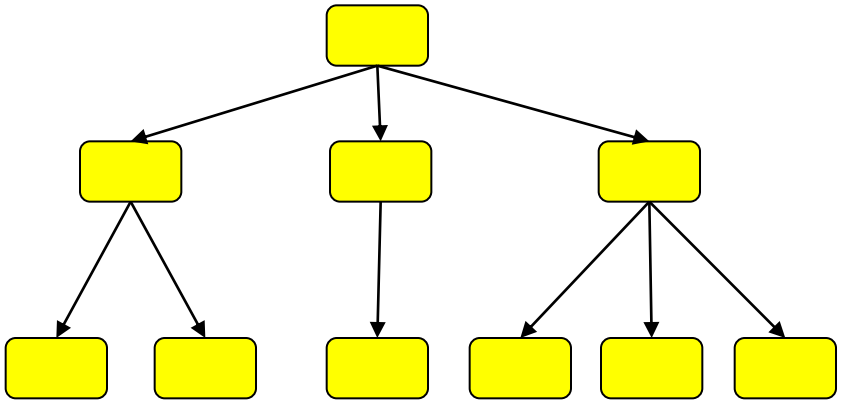
Agenda

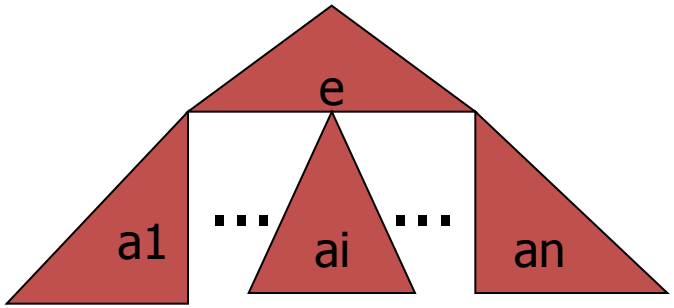
- Algorítmica de árboles n-arios
- Árboles 1-2-3
- Árboles 2-3
- Árboles B

Universidad de los Andes

Árboles n-arios

- Generalización de los árboles binarios
- Estructura recursiva en la que cada elemento tiene n árboles n-arios asociados





Formalismo abstracto

Universidad de los Andes

Árboles n-arios

- Conceptos que se extienden de árboles binarios
  - Nodo: elemento del árbol
  - Raíz: nodo inicial del árbol
  - Hoja: nodo sin hijos
  - Camino: nodos entre dos elementos incluyéndolos
  - Rama: camino entre la raíz y una hoja
  - Altura: número de nodos en la rama más larga
  - Peso: número de nodos en el árbol

Universidad de los Andes

Árboles n-arios

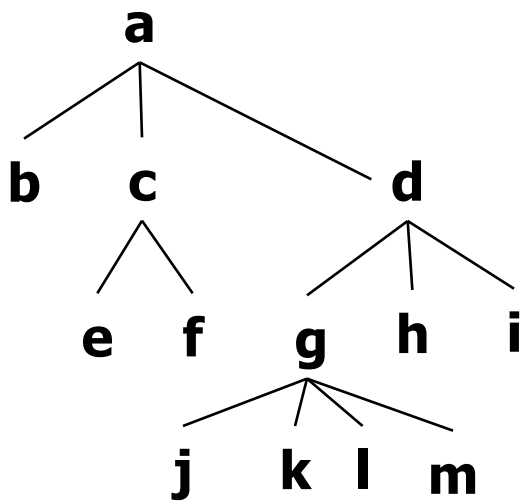
- Conceptos que se extienden de árboles binarios
  - Orden de un elemento:
    - Número de subárboles asociados
    - Una hoja es un elemento de orden 0
  - Orden de un árbol n-ario:
    - Máximo orden de sus elementos

Universidad de los Andes

Árboles n-arios

Recorridos:

- Inorden:
- Preorden:
- Postorden:



b,e,f,c,j,k,l,m,g,h,i,d,a

Universidad de los Andes

Implementación genérica

```
public class ArbolNAario<T>
{
    // -----------------------------
    // Atributos
    // -----------------------------
    /**
     * Raíz del árbol
     */
    private NodoArbolNario<T> raiz;
```

Universidad de los Andes

Implementación genérica

```
public class NodoArbolNario<T>
{
    // -----------------------------------
    // Atributos
    // -----------------------------------

    /**
     * Elemento del nodo
     */
    private T elem;

    /**
     * Subárboles
     */
    private Lista<NodoArbolNario<T>> hijos;
```

Universidad de los Andes

Algorítmica

- Similar a un árbol binario
  - Primer nivel:
    - trata el caso de un árbol vacío
    - crea estructuras de datos como iteradores
  - Segundo nivel:
    - planteamiento recursivo
    - Se tienen n avances posibles en la recursión
    - Se requiere un ciclo para iterar sobre cada avance.

Universidad de los Andes

Ejemplo

```
public int darAltura( )
{
    return ( raiz != null ) ? raiz.darAltura( ) : 0;
}
```

```
public int darAltura( )
{
    if( esHoja( ) )
        return 1;
    else
    {
        int maxAltura = 0;
        for( int i = 0; i < hijos.darLongitud( ); i++ )
        {
            NodoArbolNario<T> hijo = hijos.darElemento( i );
            int auxAltura = hijo.darAltura( );
            if( auxAltura > maxAltura )
                maxAltura = auxAltura;
        }
        return maxAltura + 1;
    }
}
```

Universidad de los Andes

Ejemplo

```
public int contarHojas( )
{
    return ( raiz != null ) ? raiz.contarHojas( ) : 0;
}
```

```
public int contarHojas( )
{
    if( esHoja( ) )
        return 1;
    else
    {
        int numHojas = 0;
        for( int i = 0; i < hijos.darLongitud( ); i++ )
        {
            numHojas += hijos.darElemento( i ).contarHojas( );
        }
        return numHojas;
    }
}
```

Universidad de los Andes

```java
public T buscar( T modelo )
{
    return ( raiz != null ) ? raiz.buscar( modelo ) : null;
}
```

```java
public T buscar( T modelo )
{
    if( modelo.equals( elem ) )
        return elem;
    else if( esHoja( ) )
        return null;
    else
    {
        for( int i = 0; i < hijos.darLongitud( ); i++ )
        {
            T aux = hijos.darElemento( i ).buscar( modelo );
            if( aux != null )
                return aux;
        }
        return null;
    }
}
```

Universidad de los Andes

A trabajar...

```
public class ArbolNArio<T>
{

   public Iterador<T> darNivel( int nivel )
   {









   }
}
```

Universidad de los Andes

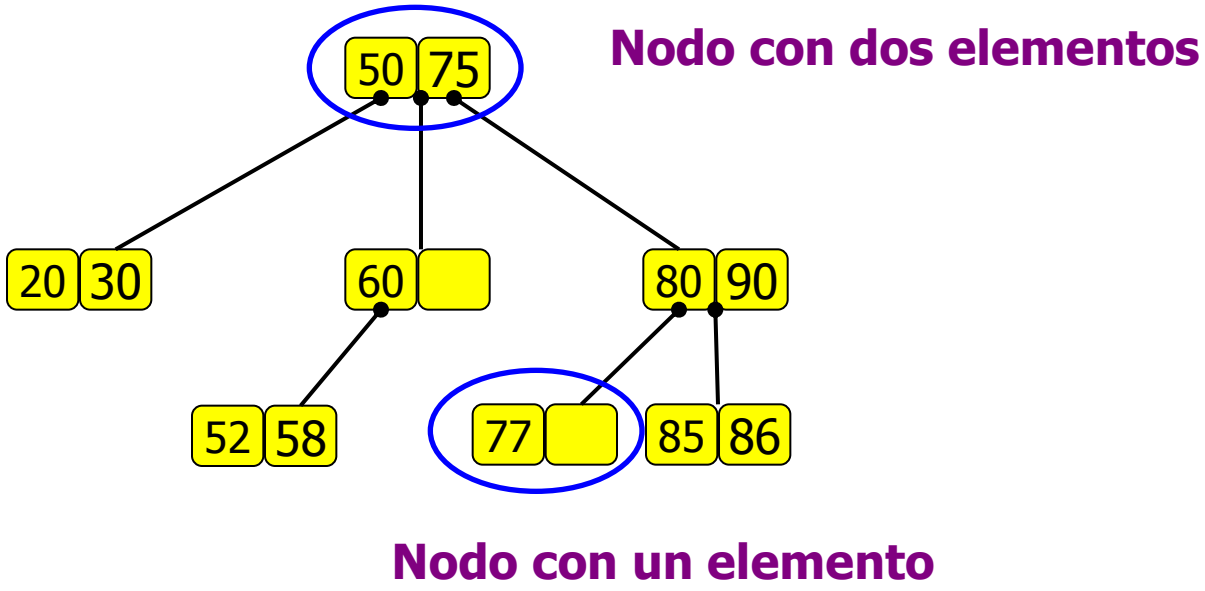
Agenda

- Algorítmica de árboles n-arios
- Árboles 1-2-3
- Árboles 2-3
- Árboles B

Universidad de los Andes

Árbol 1-2-3: Árbol triario ordenado

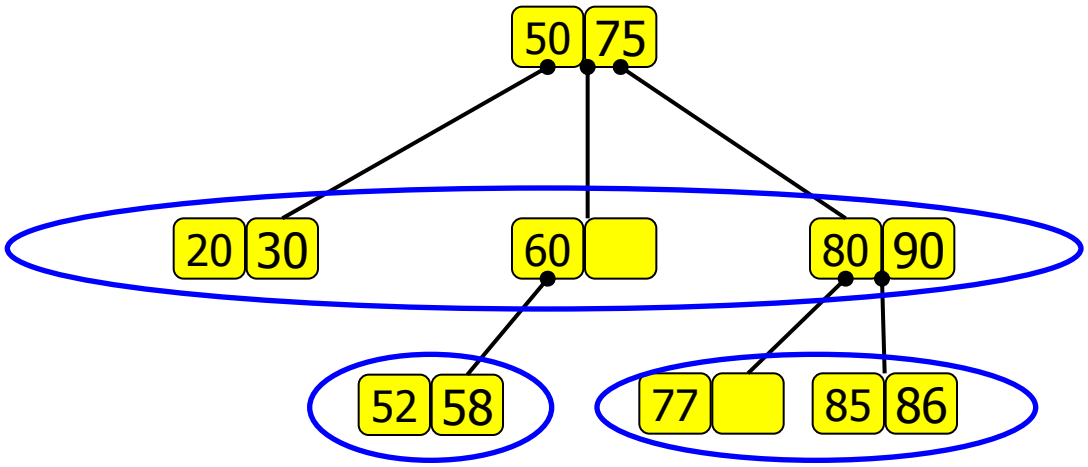
- Árbol n-ario ordenado de orden 3
- En cada nodo tiene 1 ó 2 elementos



Universidad de los Andes

Árbol 1-2-3: Árbol triario ordenado

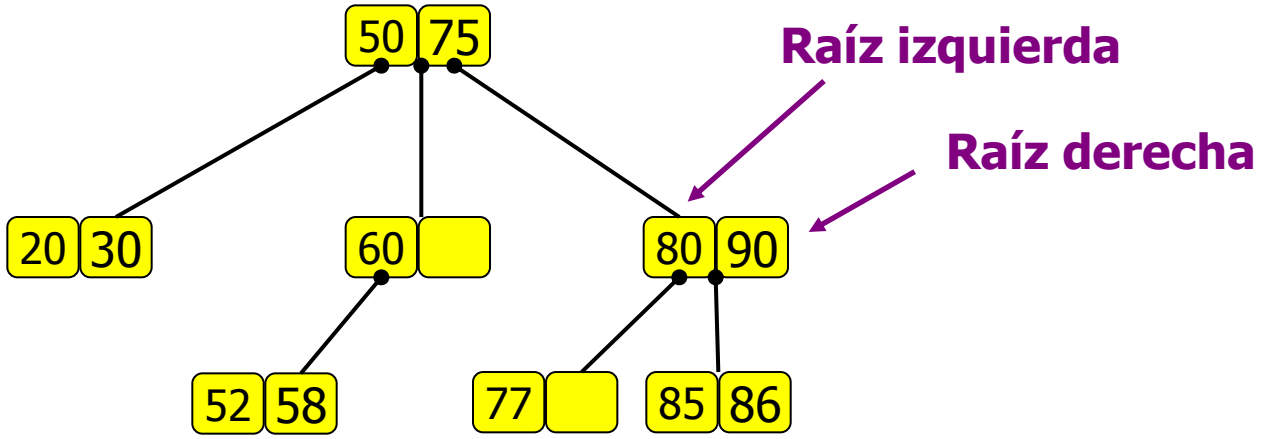
- Cada nodo tiene 1, 2 ó 3 subárboles asociados



Universidad de los Andes

Árbol 1-2-3: Árbol triario ordenado

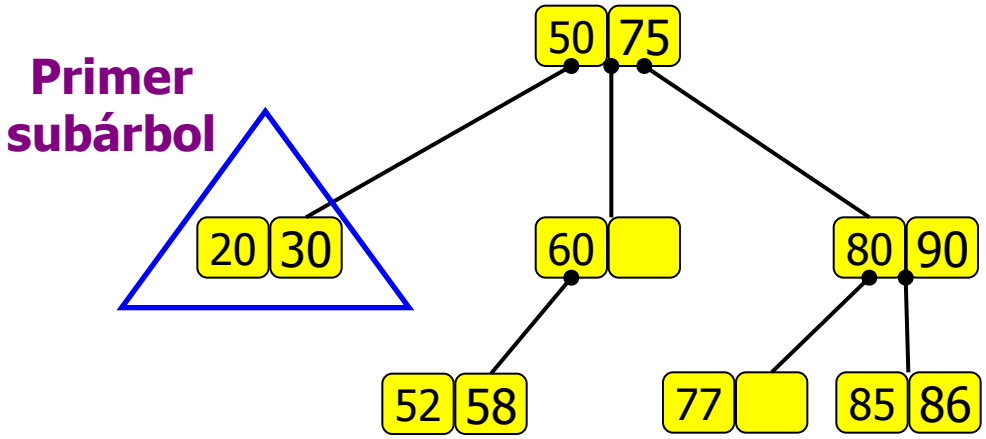
- No hay elementos repetidos
- El elemento de la izquierda de cada nodo (raíz izquierda) es menor que el elemento de su derecha (raíz derecha)



Universidad de los Andes

Árbol 1-2-3: Árbol triario ordenado

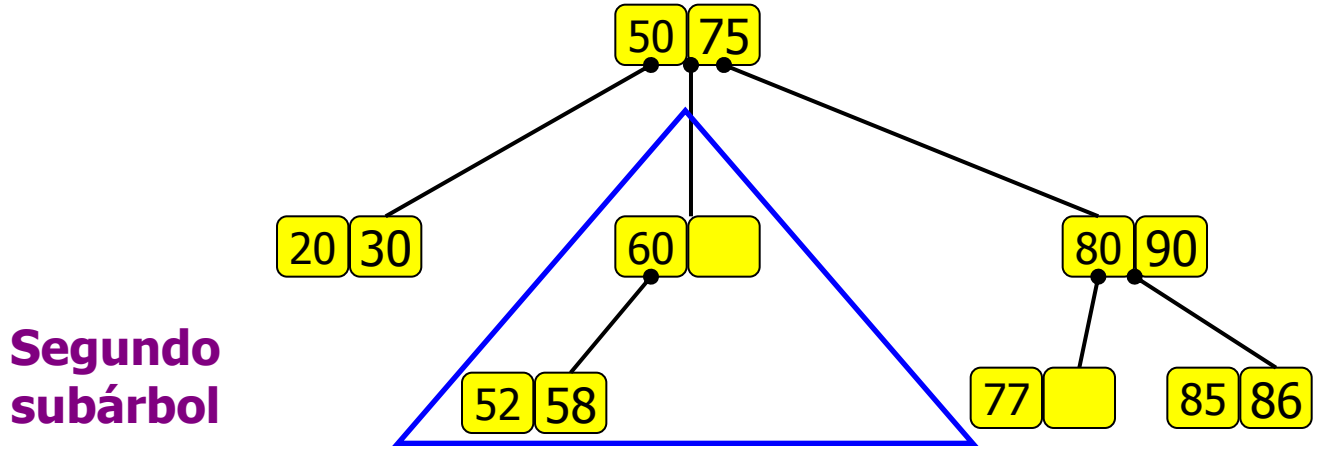
- El primer subárbol es un árbol 1-2-3 que contiene elementos menores que la raíz izquierda



Universidad de los Andes

Árbol 1-2-3: Árbol triario ordenado

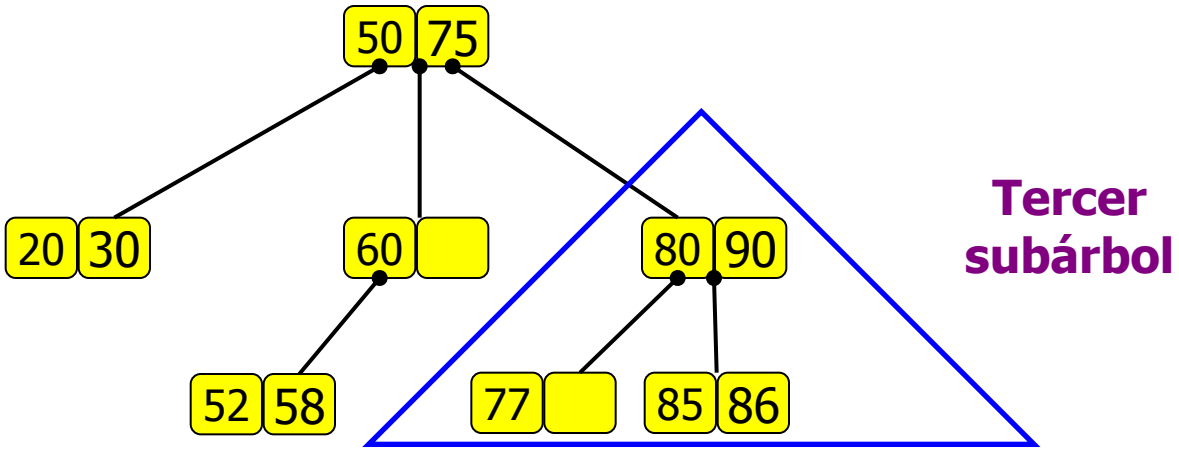
- El segundo subárbol es un árbol 1-2-3 que contiene elementos mayores que la raíz izquierda pero menores que la raíz derecha



Universidad de los Andes

Árbol 1-2-3: Árbol triario ordenado

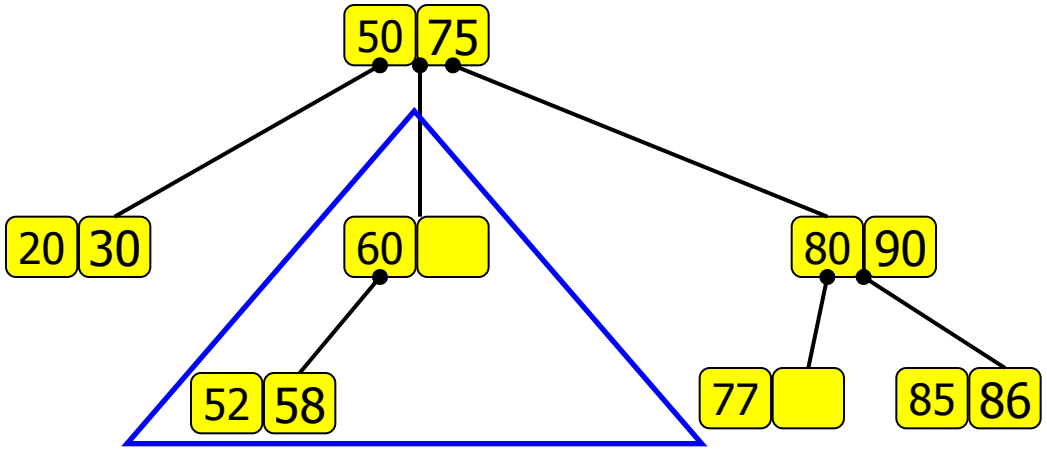
- El tercer subárbol es un árbol 1-2-3 que contiene los elementos mayores que la raíz derecha



Universidad de los Andes

Árbol 1-2-3: Árbol triario ordenado

- Si la raíz derecha está vacía, su tercer subárbol debe ser vacío (el segundo puede o no ser vacío).



Universidad de los Andes

A trabajar...

- Declaración en Java de la clase
  - Para valores enteros simples

- Algorítmica básica:
  - Búsqueda
  - Inserción
  - Supresión

Universidad de los Andes

Agenda

- Algorítmica de árboles n-arios
- Árboles 1-2-3
- Árboles 2-3
- Árboles B

Universidad de los Andes

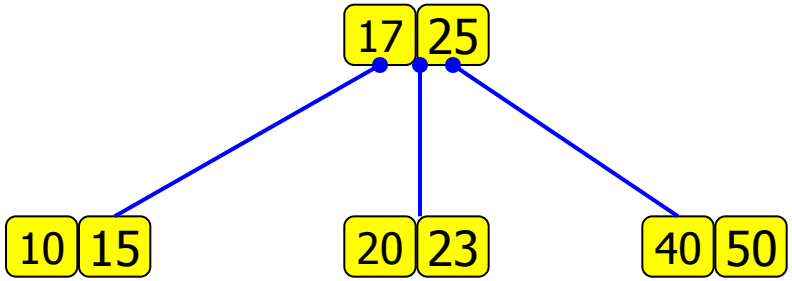
Árbol 2-3: Árbol triario ordenado balanceado

- Motivación:
  - Optimizar el tiempo de acceso en una estructura de datos en memoria secundaria
  - Acceso a la información en O(log3N)
  - Baja complejidad en los algoritmos de actualización.

Universidad de los Andes

Árboles 2-3

- Es un árbol 1-2-3 y además:
  - Todas las hojas se encuentran en el mismo nivel
  - Todos los nodos internos tienen por lo menos 2 subárboles asociados no vacíos, aunque la raíz derecha este vacía.



Universidad de los Andes

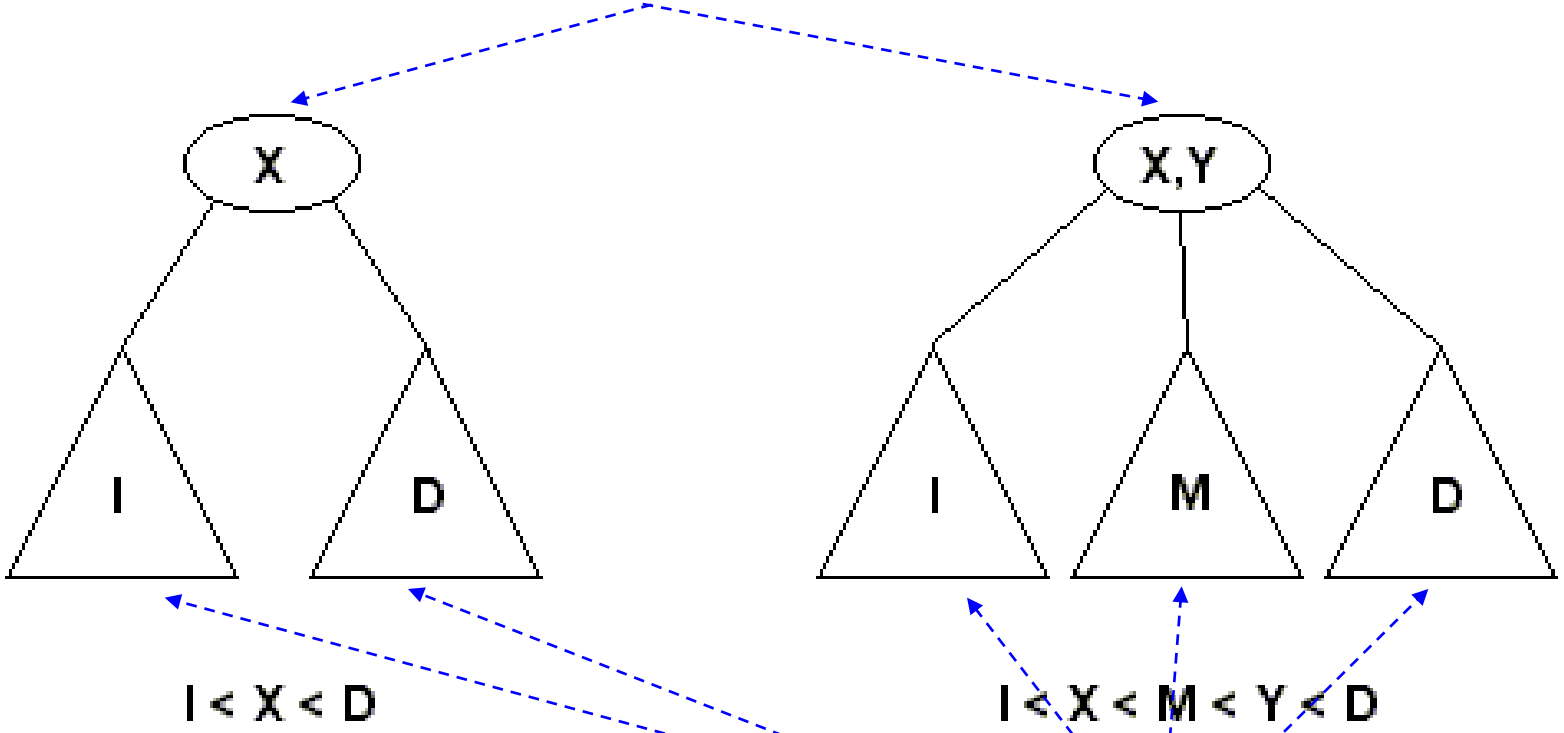
Árbol 2-3

- Todos los nodos pueden tener hasta 2 elementos.
- Un nodo interno puede tener 2 ó 3 hijos, dependiendo de cuántos elementos posea el nodo:
  - Si hay 1 elemento en el nodo, debe tener 2 hijos
  - Si hay 2 elementos en el nodo, debe tener 3 hijos

Universidad de los Andes

Ejemplos

Cada nodo tiene hasta 2 elementos

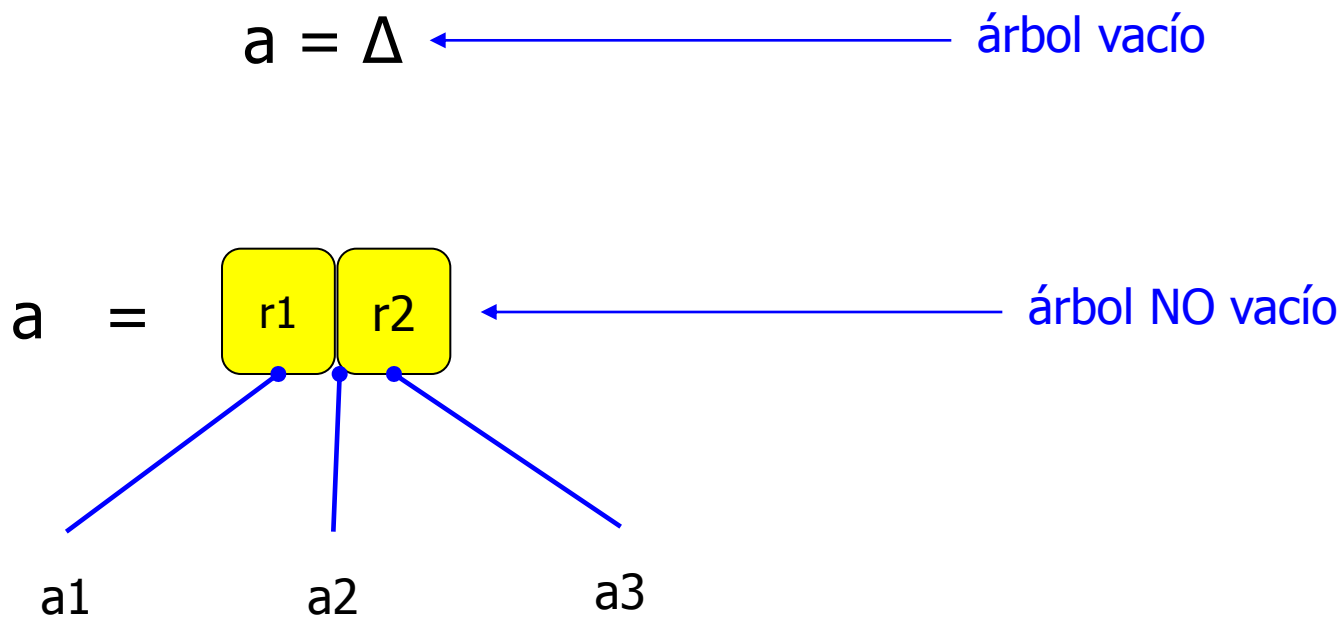


Cada nodo interno puede tener 2 o 3 hijos (dependiendo de cuántos elementos posea el nodo)

Universidad de los Andes

Árbol 2-3

- Formalismo abstracto



Universidad de los Andes

Árbol 2-3: Inserción

- El crecimiento NO se hace a nivel de las hojas
  - Aunque la inserción se sigue haciendo en las hojas
- El crecimiento se hace a nivel de la raíz
  - Todas las hojas se deben mantener siempre en el mismo nivel

Universidad de los Andes

Árbol 2-3: Inserción

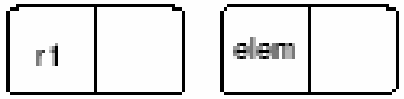
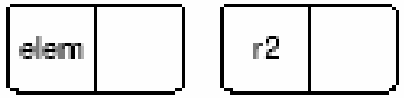
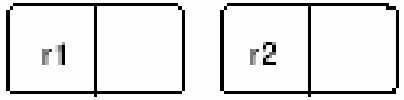
1. Localizar la hoja en la cual se debe agregar el elemento
2. Insertar
   - Caso1: Existe espacio en el nodo -> la estructura del árbol NO se altera

Situación inicial	Situación final
 , r1 < elem	
 , r1 > elem	

Universidad de los Andes

Árbol 2-3: Inserción

- Caso 2: El nodo está lleno. Se debe modificar la estructura del árbol:
  - El nodo se parte en dos nodos del mismo nivel
  - Los tres elementos (dos elementos del nodo y el nuevo elemento) se reparten de la siguiente manera:

Situación inicial	Situación intermedia
 , elem > r2	 , sube r2
 , elem < r1	 , sube r1
 , r1 < elem < r2	 , sube elem

Universidad de los Andes

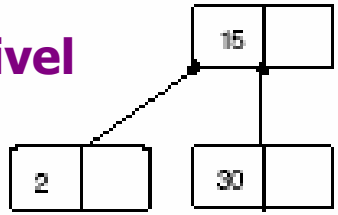
Árbol 2-3: Inserción

Situación inicial	Situación intermedia
r1 | r2 , elem > r2	r1 , elem , sube r2
r1 | r2 , elem < r1	elem , r2 , sube r1
r1 | r2 , r1 < elem < r2	r1 , r2 , sube elem

- El elemento que no fue incluido en los dos nodos nuevos se sube en la estructura y se inserta en su padre.
- Se repite el proceso hacia arriba:
  - Al partir en dos el nodo se está generando un nuevo subárbol que puede generar que los ancestros se tengan que partir a su vez para poderlo incluir.

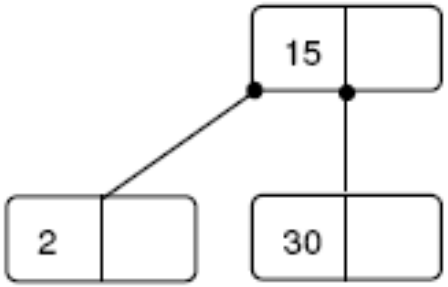
Universidad de los Andes

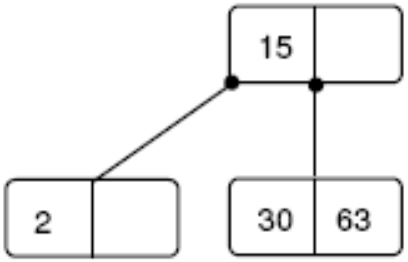
Árbol 2-3 : Inserción

	Insertar el elemento 30: se crea una hoja y se coloca el elemento como raiz izquierda
 **Se parte en dos**	Insertar el elemento 2: corresponde al caso 1. Se mueve a la derecha la raiz izquierda para dar cabida al nuevo elemento.
Se crea un nivel 	Insertar el elemento 15: corresponde al caso 2. Encuentra una hoja llena. La parte en dos nodos e inserta en el padre el elemento que se encuentre en la mitad de los tres (2 < 15 < 30). Como el padre es vacio, se crea un nuevo nivel, se coloca el elemento como la raiz izquierda del nodo, y se le asocian los dos nodos que se acaban de partir.

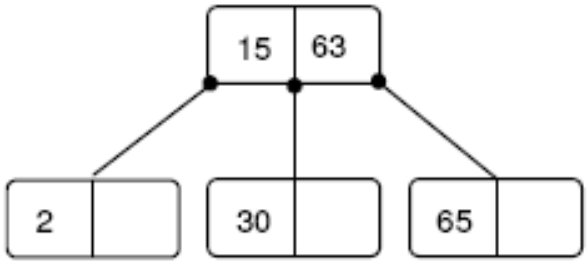
Universidad de los Andes

Árbol 2-3 : Inserción





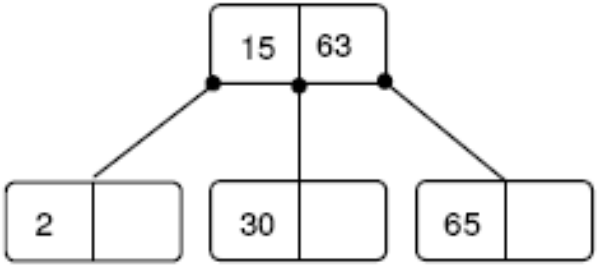
Insertar el elemento 63: corresponde al caso 1

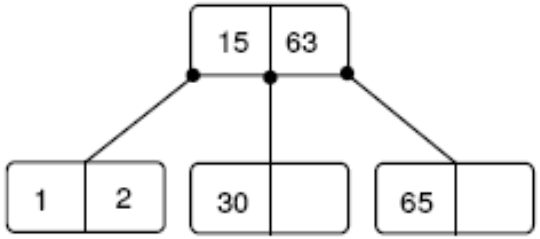


Insertar el elemento 65: corresponde al caso 2. Se parte la hoja y sube al padre el elemento de la mitad (63). Al insertar dicho valor en el padre se trata como el caso 1, porque en el nodo hay espacio.

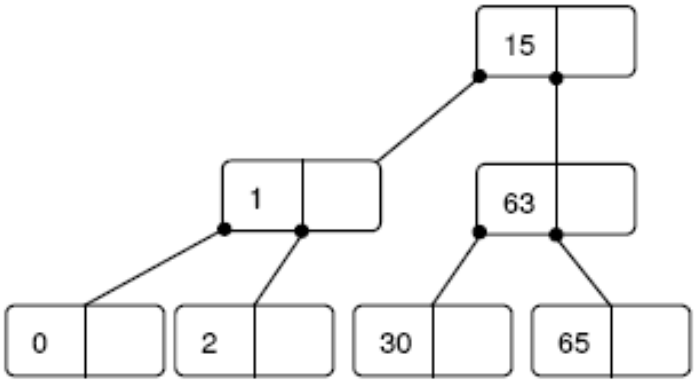
Universidad de los Andes

Árbol 2-3 : Inserción





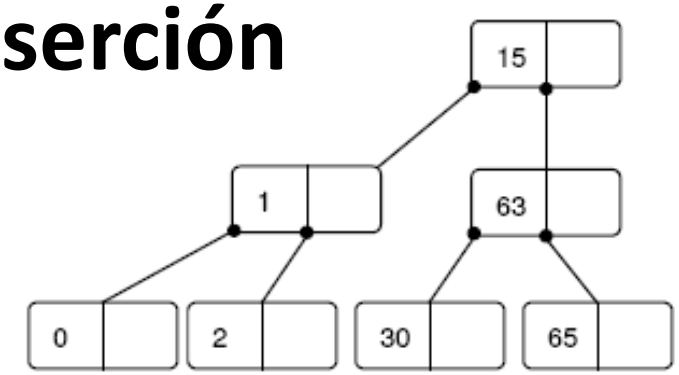
Insertar el elemento 1: corresponde al caso 1.

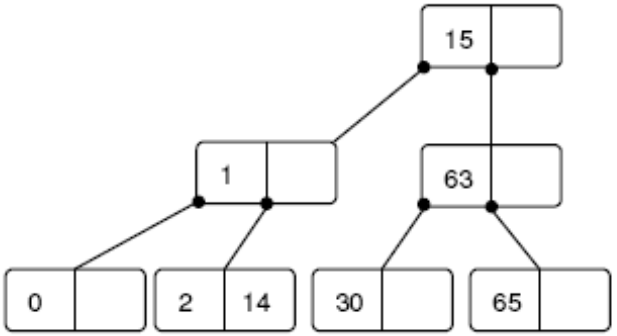


Insertar el elemento 0: corresponde al caso 2. Se parte la hoja [1, 2], se colocan allí los elementos 0 y 2, y sube el valor 1 a su padre. Como el nodo del padre [15, 63] está lleno también se debe partir, dejando en ese nivel los elementos 1 y 63, y subiendo el 15.

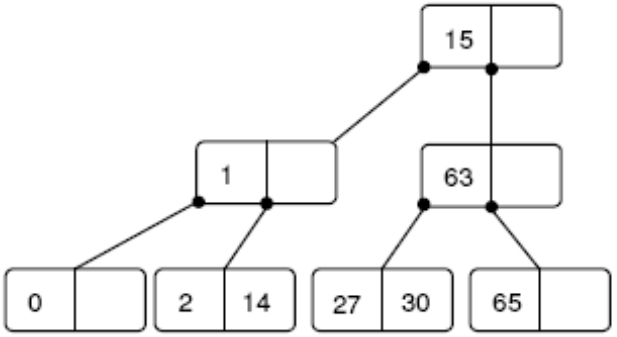
Universidad de los Andes

Árbol 2-3 : Inserción





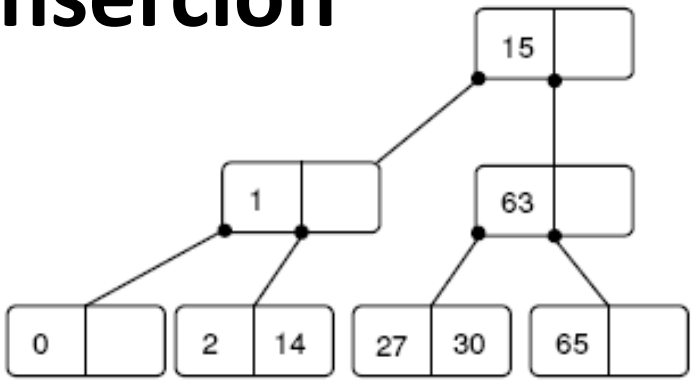
Insertar el elemento 14: corresponde al caso 1.

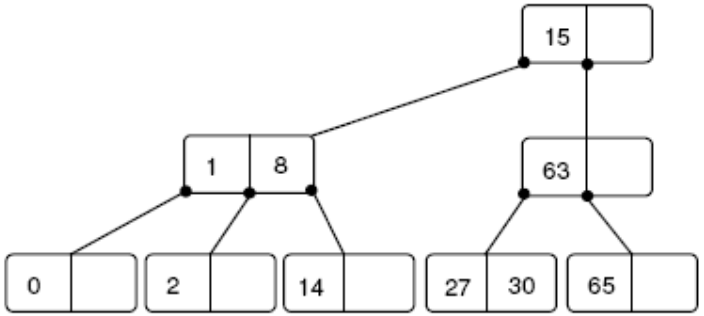


Insertar el elemento 27: corresponde al caso 1.

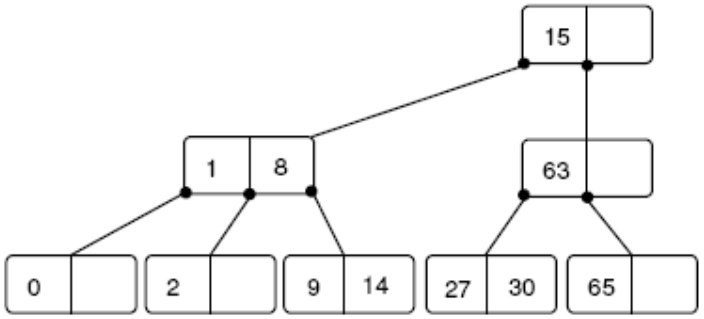
Universidad de los Andes

Árbol 2-3 : Inserción





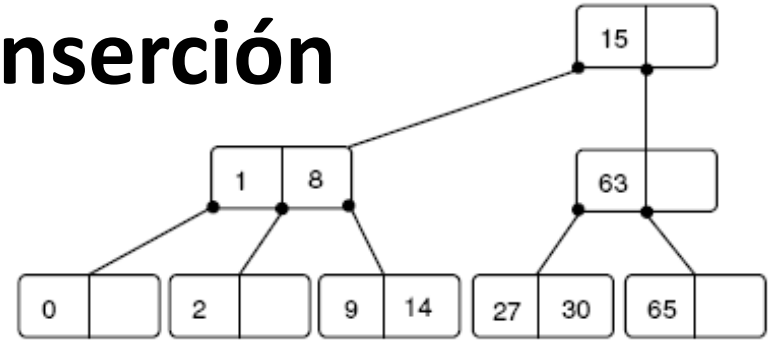
Insertar el elemento 8: corresponde al caso 2. Se parte el nodo [2, 14] y sube el 8. Allí encuentra espacio y se coloca como raíz derecha.

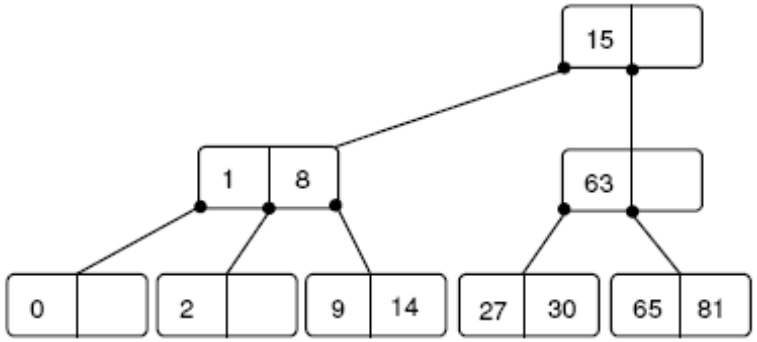


Insertar el elemento 9: corresponde al caso 1.

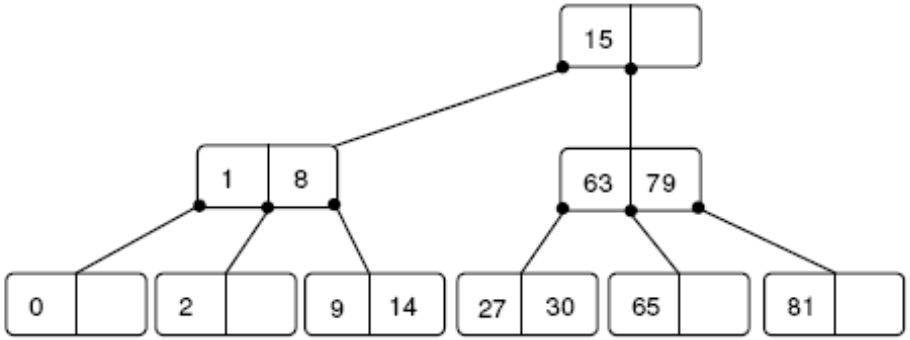
Universidad de los Andes

Árbol 2-3 : Inserción





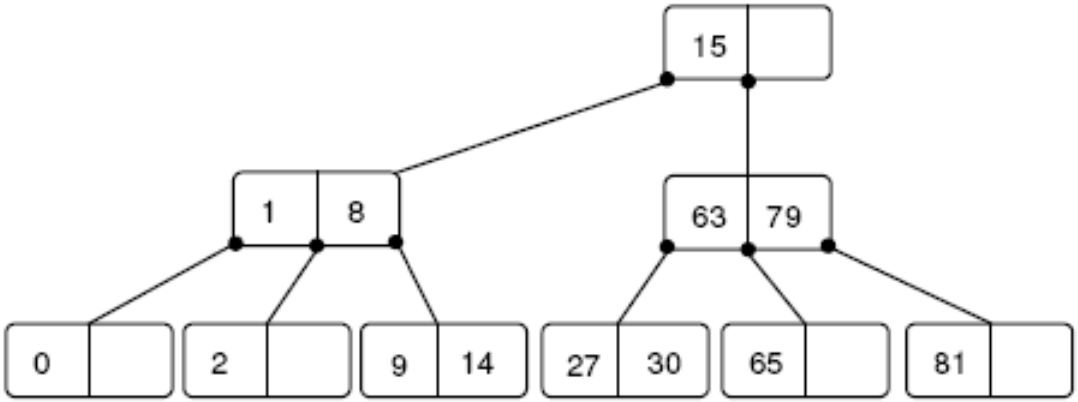
Insertar el elemento 81: corresponde al caso 1.

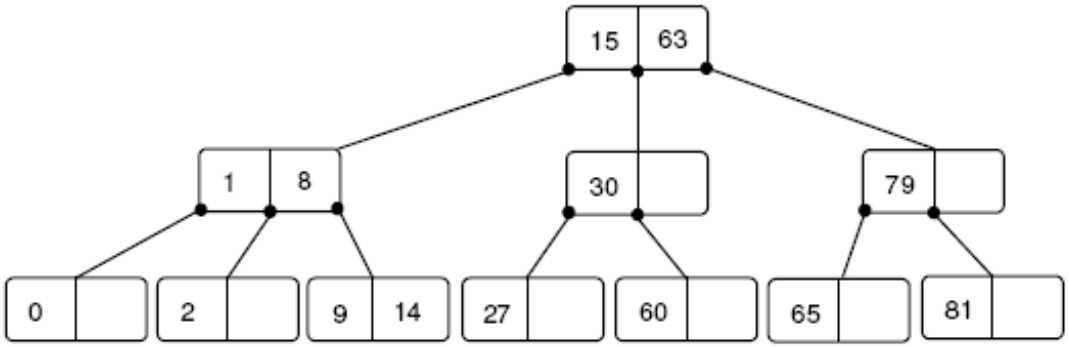


Insertar el elemento 79: corresponde al caso 2. Se parte el nodo [65, 81] y sube el elemento 79.

Universidad de los Andes

Árbol 2-3 : Inserción



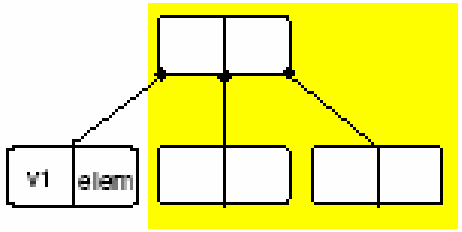
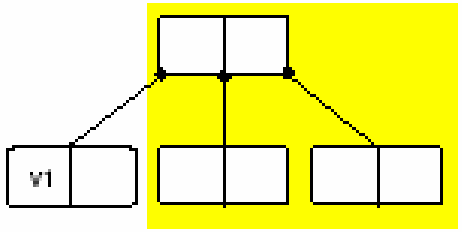
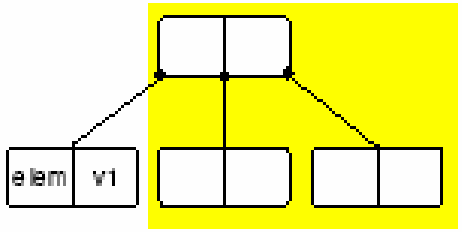
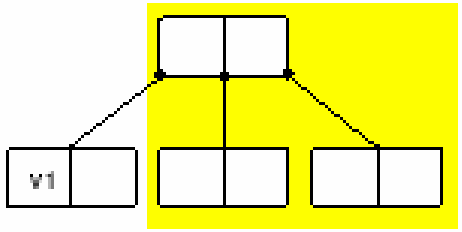
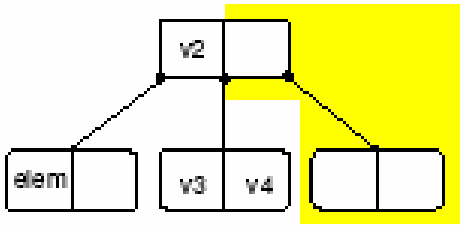
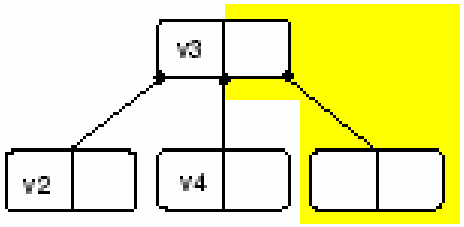


Insertar el elemento 60: corresponde al caso 2. Se parte el nodo [27, 30], se incluye el 60 y sube el elemento 30. Como su padre está lleno se parte en los nodos [30] y [79], y sube el elemento 63. Este elemento se sitúa en la raíz derecha del árbol, donde hay espacio libre.

Universidad de los Andes

Árbol 2-3: Eliminación

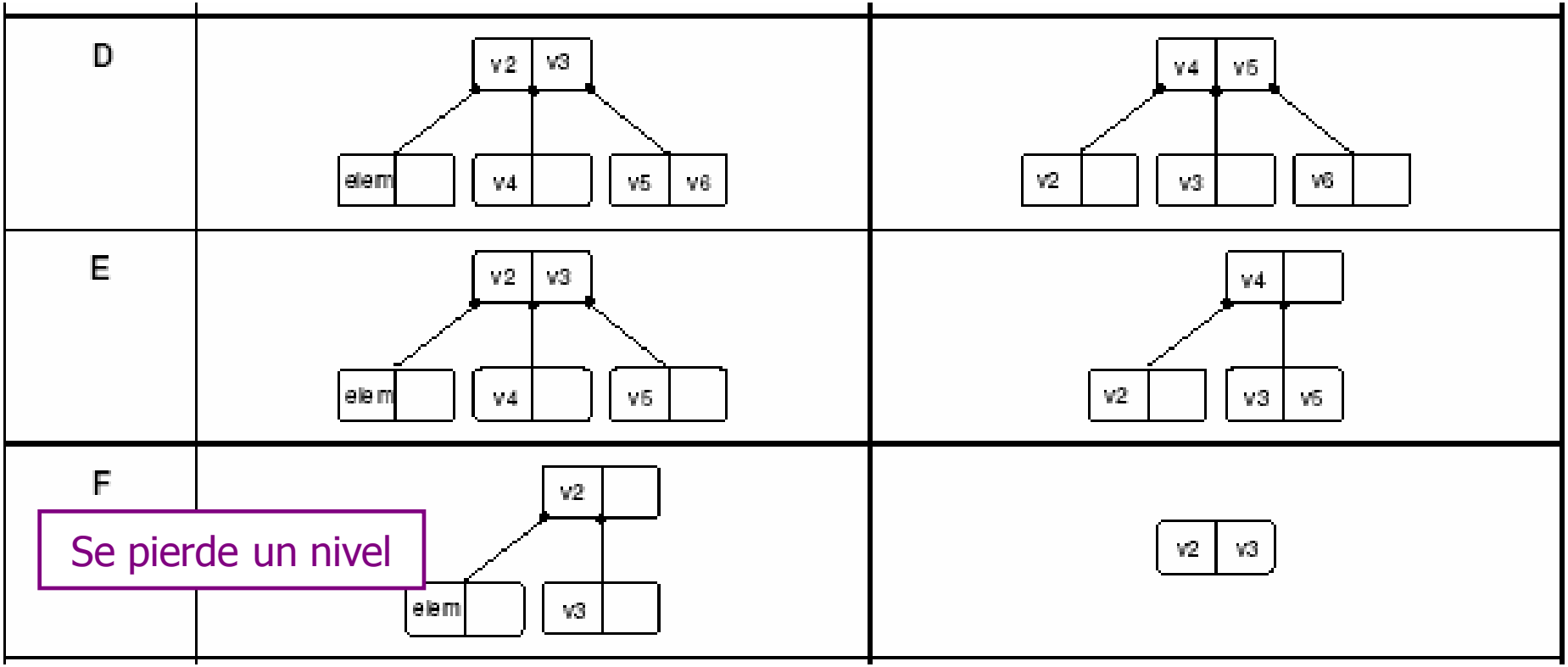
- Caso1: El elemento está en una hoja

Caso	Situación inicial	Solución
A		
B		
C		

Universidad de los Andes

Árbol 2-3 : Eliminación

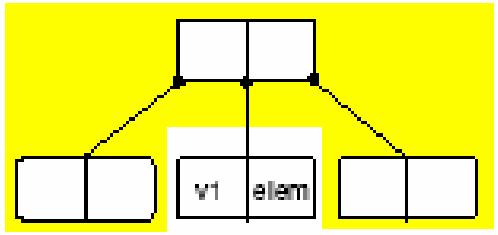
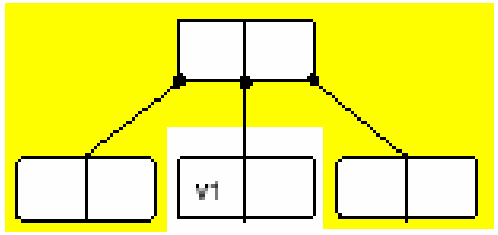
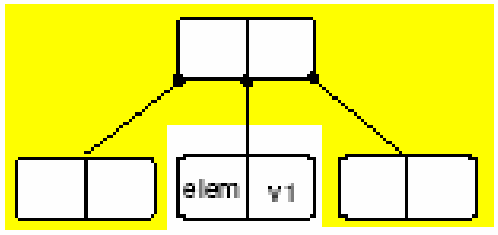
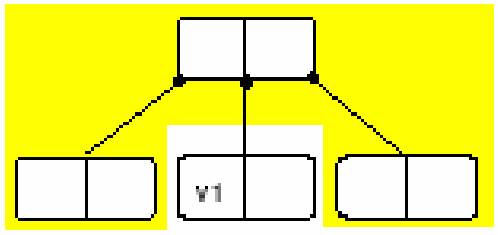
- Caso1: El elemento está en una hoja



Universidad de los Andes

Árbol 2-3 : Eliminación

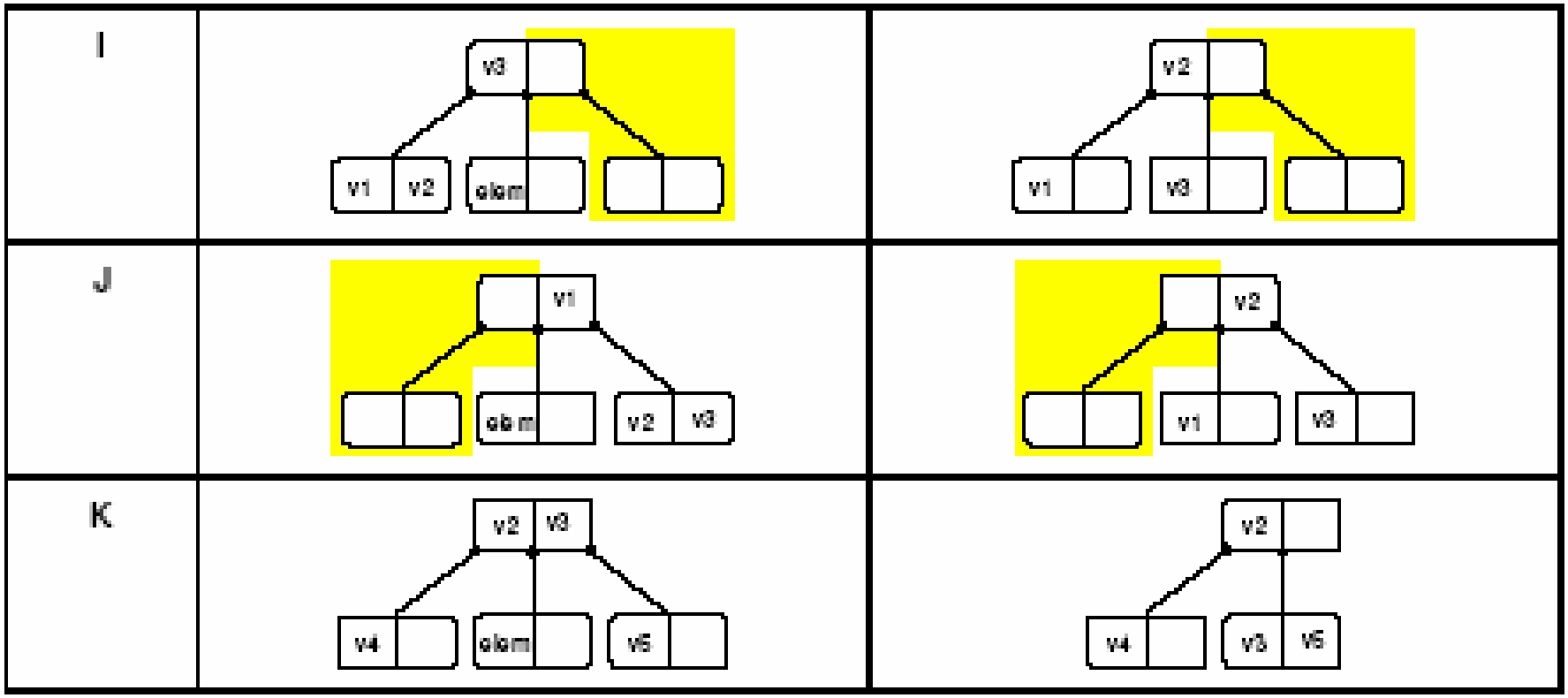
- Caso1: El elemento está en una hoja

G		
H		

Universidad de los Andes

Árbol 2-3 : Eliminación

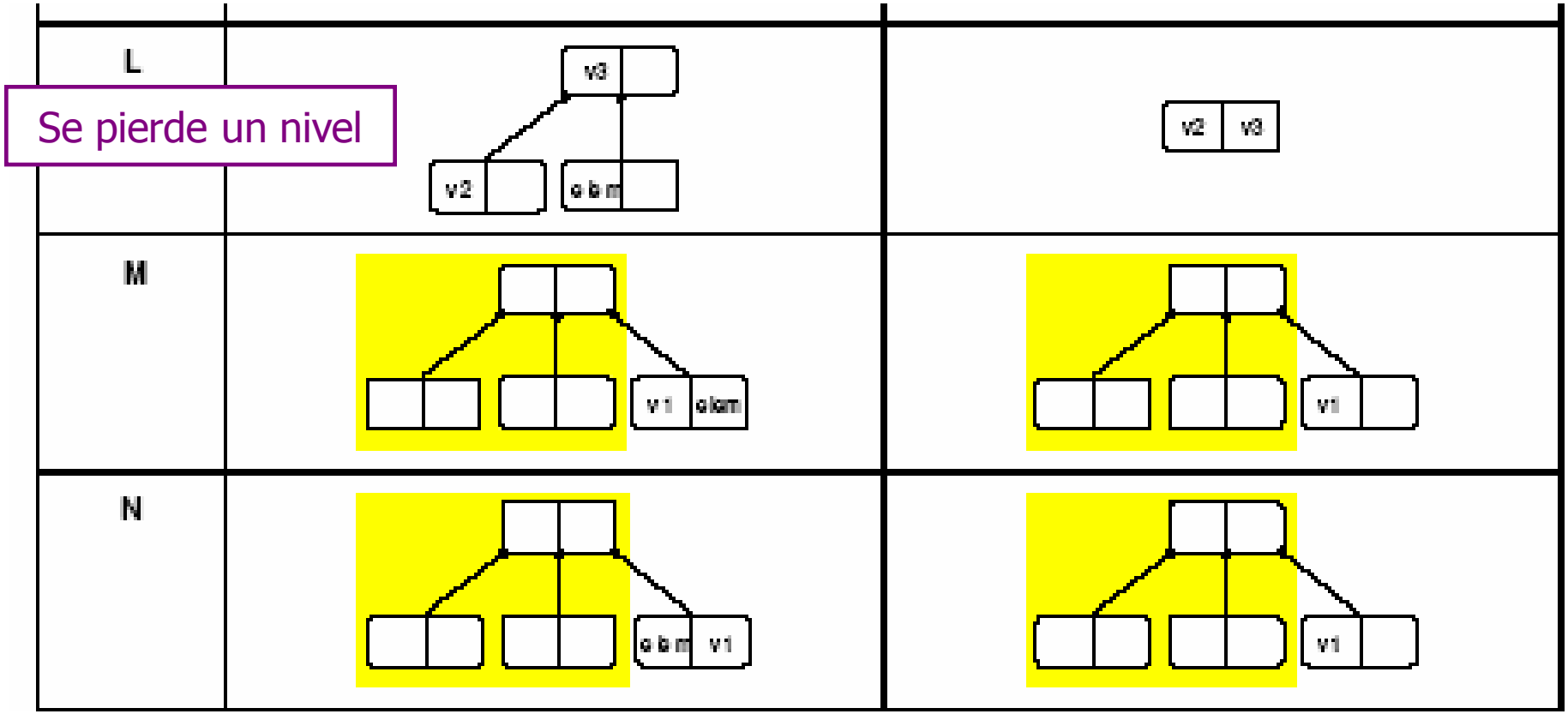
- Caso1: El elemento está en una hoja



Universidad de los Andes

Árbol 2-3 : Eliminación

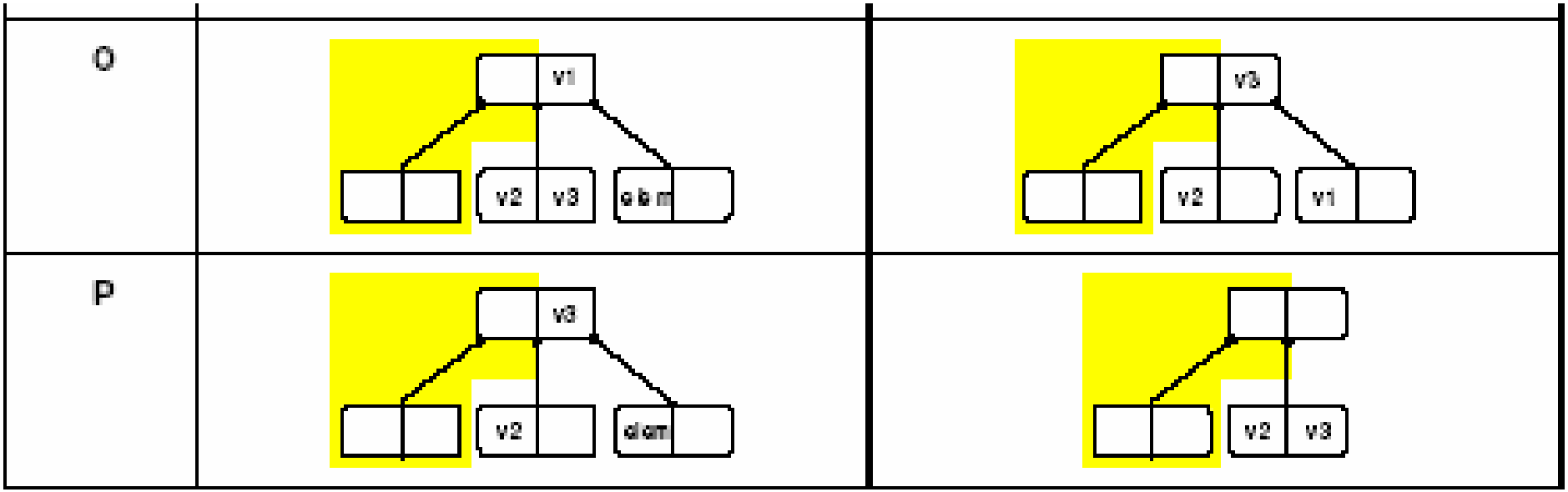
- Caso1: El elemento está en una hoja



Universidad de los Andes

Árbol 2-3 : Eliminación

- Caso1: El elemento está en una hoja



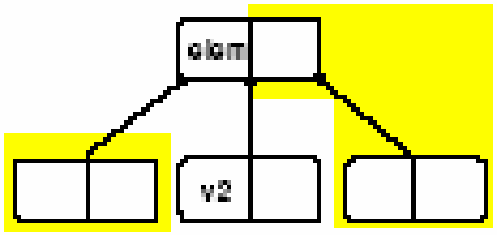
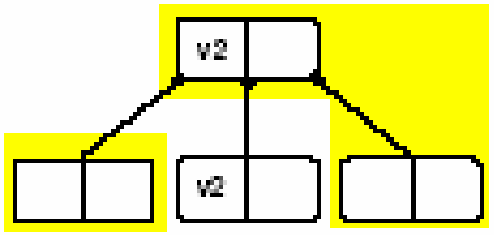
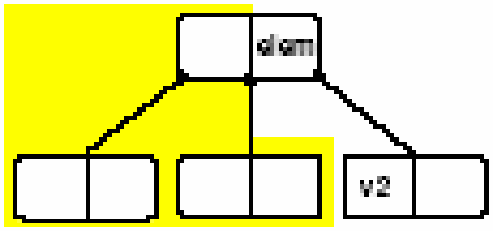
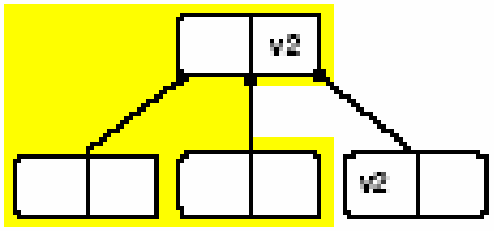
Universidad de los Andes

Árbol 2-3 : Eliminación

- Caso 2: El elemento no está en una hoja
  - Se busca un valor que se encuentre en una hoja y que pueda reemplazar el valor. (Como está ordenado, el candidato sería el menor del subárbol derecho)
    - Solución para el caso 1

Universidad de los Andes

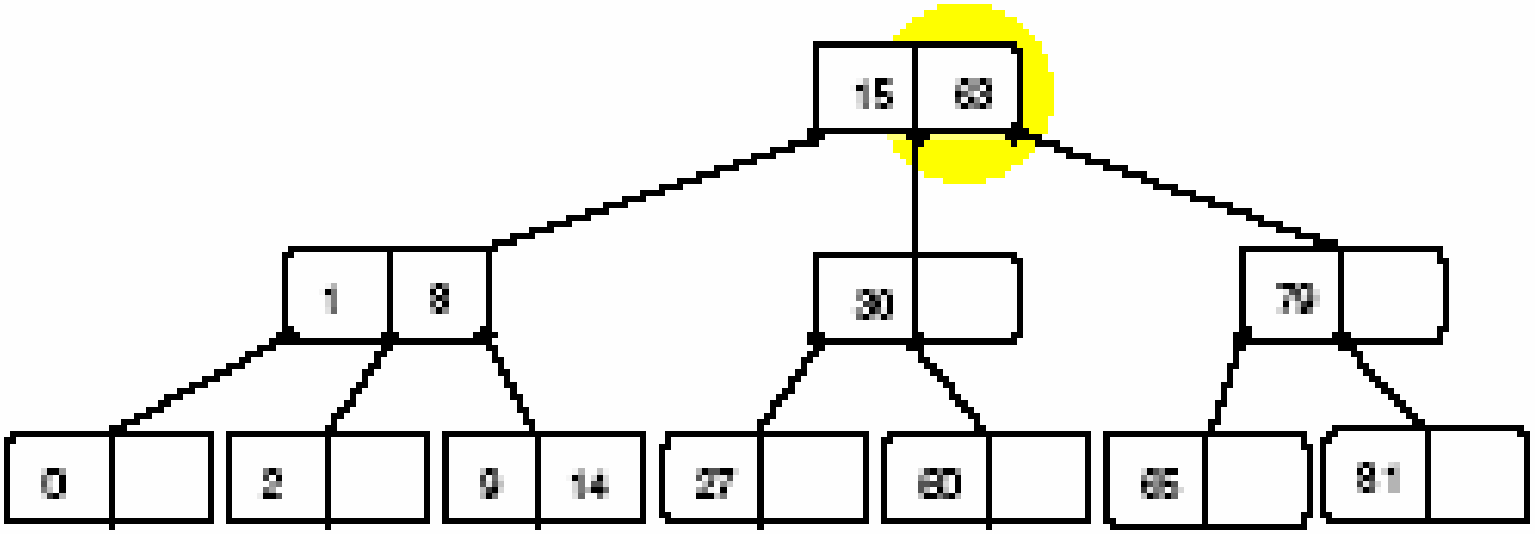
Árbol 2-3 : Eliminación

	Situación inicial	Situación intermedia: Eliminar v2 de una hoja (Caso 1)
Q		
R		

Universidad de los Andes

Árbol 2-3 : Eliminación

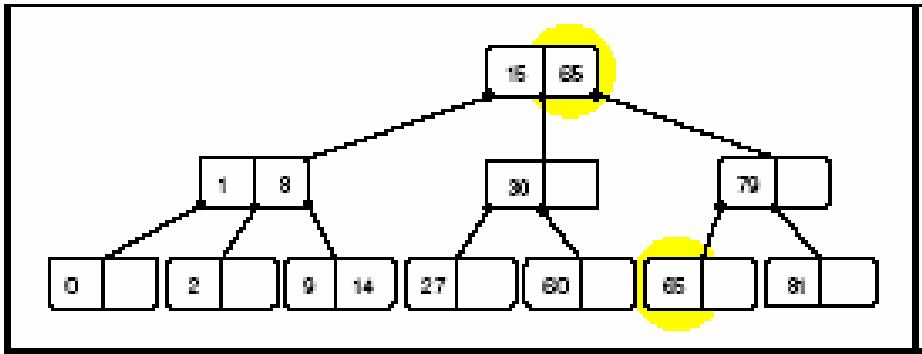
- Ejemplo: Eliminar el 63



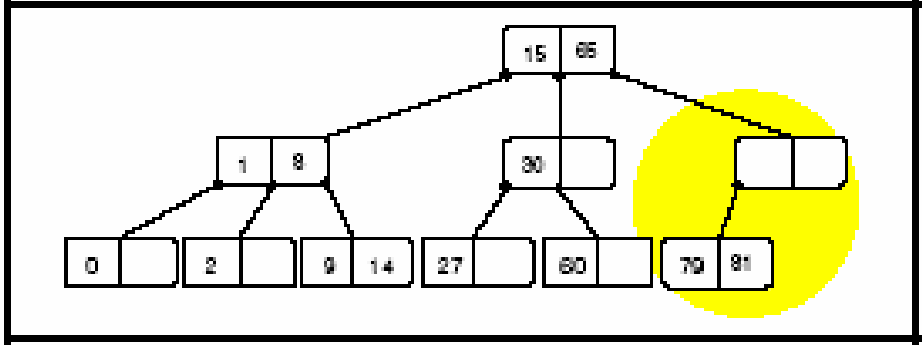
Universidad de los Andes

Árbol 2-3 : Eliminación

- Ejemplo: Eliminar el 63



(1) Se busca un valor en una hoja que pueda ocupar el espacio que va a liberar el 63. Se escoge para esto el elemento 65 (caso 2-R), y se replantea el problema para eliminar este valor.

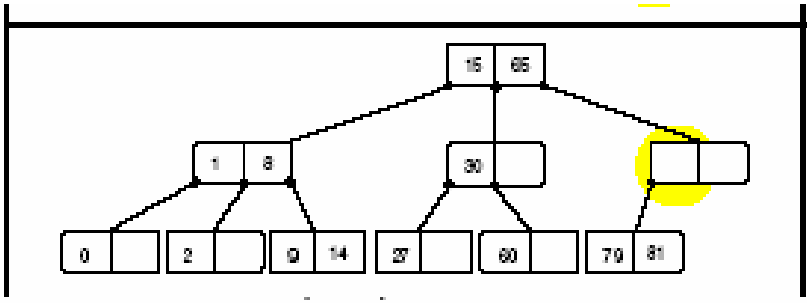


(2) El tercer subárbol entra en el caso 1-F, en el cual se pierde un nivel. Se llega al siguiente caso intermedio, porque el árbol debe decrecer por la raíz.

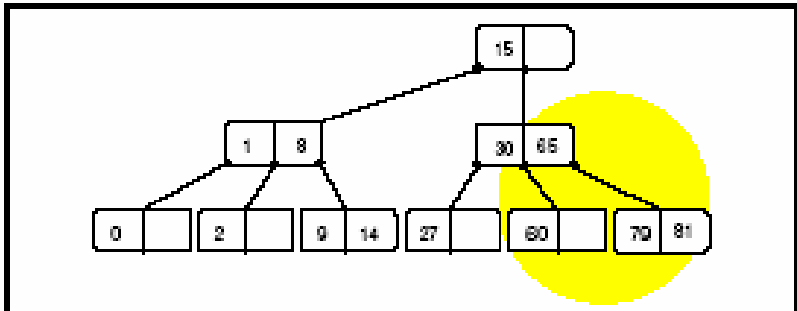
Universidad de los Andes

Árbol 2-3 : Eliminación

- Ejemplo: Eliminar el 63



(3) El árbol completo llega al caso 1-P, en el cual el segundo subárbol no tiene raíz derecha para prestársela al tercer subárbol. Se debe aplicar la solución planteada para ese caso, pero teniendo en cuenta que el elemento no es una hoja, y que también deben moverse los subárboles asociados.



(4) El proceso termina porque el árbol ya es estructuralmente correcto.

Universidad de los Andes

Implementación

```
public class Arbol2_3<T extends Comparable<? super T>>
{
    // -----------------------------------------------------
    // Atributos
    // -----------------------------------------------------
    /**
     * Raíz del árbol 2-3
     */
    private Nodo2_3<T> raiz;
```

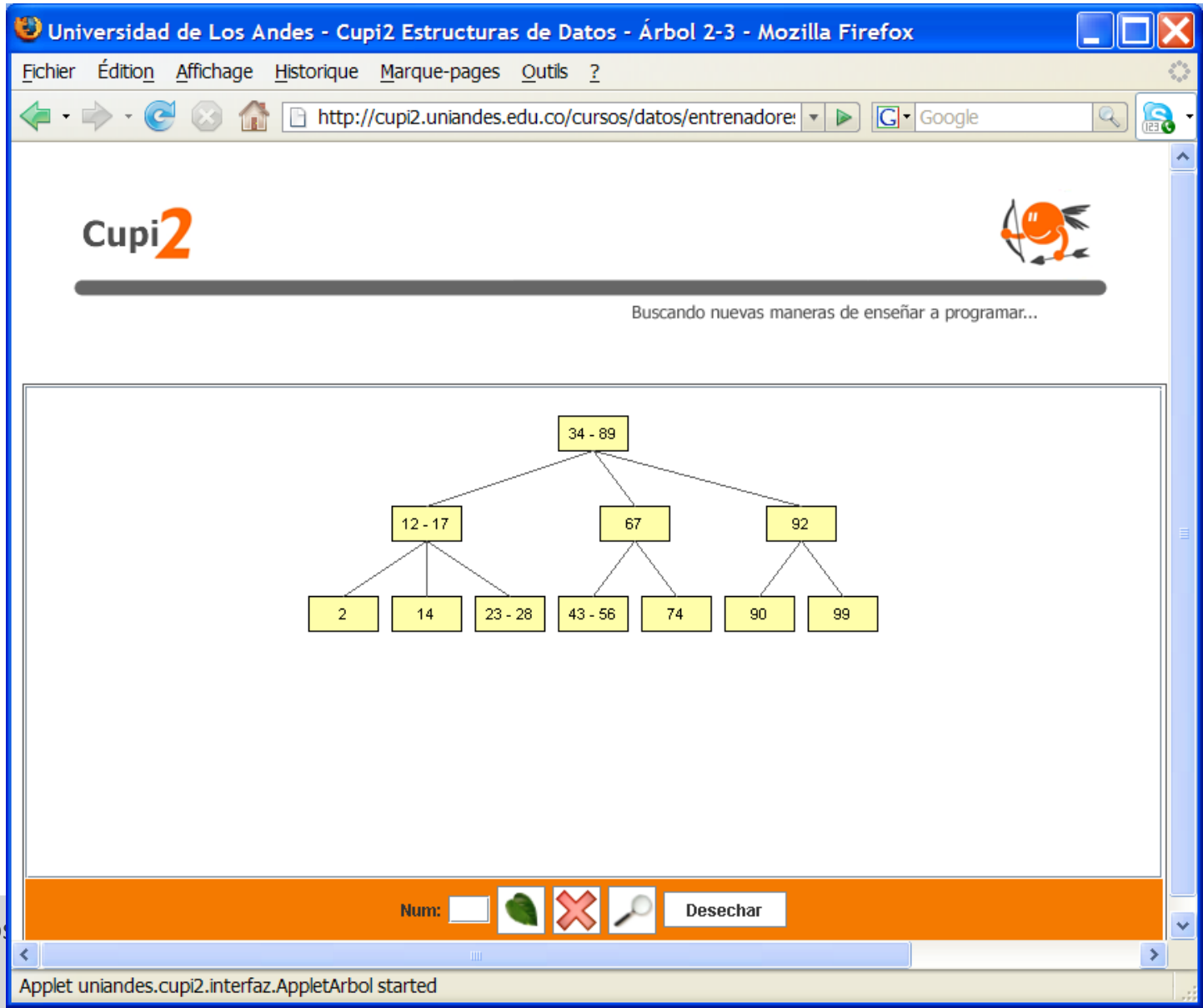
Universidad de los Andes

Implementación

```
public class Nodo2_3<T extends Comparable<? super T>>
{
    // -----------------------------------------------
    // Atributos
    // -----------------------------------------------
    /**
     * Primera raíz del nodo
     */
    private T r1;

    /**
     * Segunda raíz del nodo
     */
    private T r2;

    /**
     * Subárbol 1
     */
    private Nodo2_3<T> h1;

    /**
     * Subárbol 2
     */
    private Nodo2_3<T> h2;

    /**
     * Subárbol 3
     */
    private Nodo2_3<T> h3;
```

Universidad de los Andes

Entrenador WEB



Universidad de los Andes

A trabajar...

- Desarrolle el método que permite buscar un elemento en un árbol 2-3

```
public T buscar( T modelo )
{
    return ( raiz != null ) ? raiz.buscar( modelo ) : null;
}
```

```
public T buscar( T modelo )
{
    ...
```

Universidad de los Andes

A trabajar...

- Muestre el proceso de inserción en un árbol 2-3 de la siguiente secuencia de valores:

  - 25 – 86 – 34 – 23 – 4 – 98 – 12 – 56 – 74 – 77 - 80

Universidad de los Andes

Agenda

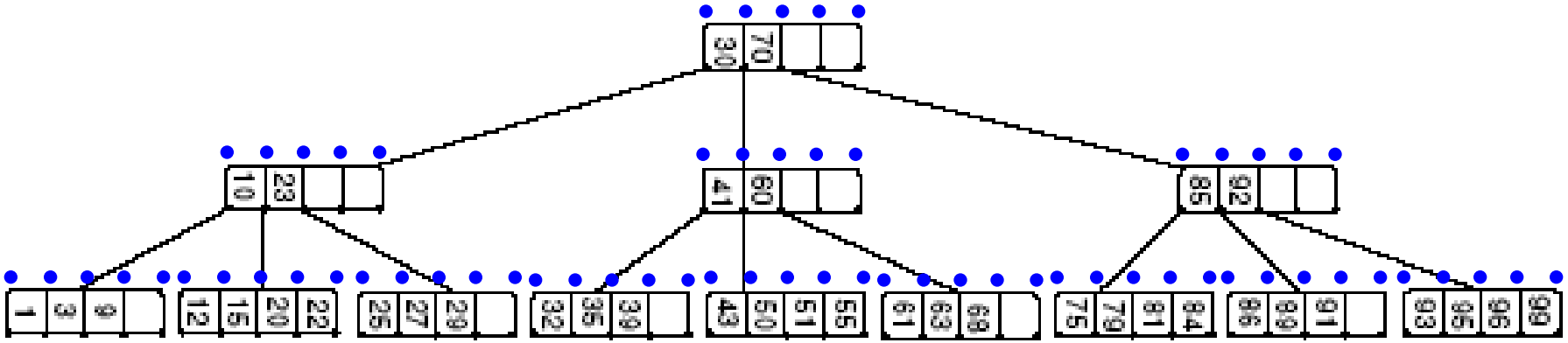
- Algorítmica de árboles n-arios
- Árboles 1-2-3
- Árboles 2-3
- Árboles B

Universidad de los Andes

Árboles B

- Árbol n-ario ordenado y balanceado
- Generalización de un árbol 2-3

B de orden 5



Universidad de los Andes

Árboles B

- Cada nodo tiene k elementos y k+1 subárboles B asociados

- Un árbol 2-3 es un árbol B de orden 3

Universidad de los Andes

Árboles B

- Para un árbol B de orden k:
  - Todas las hojas se encuentran al mismo nivel
  - Todos los nodos internos, excepto la raíz , tienen por lo menos (k+1)/2 subárboles asociados no vacíos
  - El número de elementos de un nodo interno es uno menos que el número de subárboles asociados

Universidad de los Andes

Preguntas...