Estructuras de Datos y Algoritmos

Tema 4: Árboles

Departamento de Informática
Universidad de Valladolid

Curso 2011-12

Grado en Ingeniería Informática
Grado en Ingeniería Informática de Sistemas

1. DEFINICIONES Y PROPIEDADES

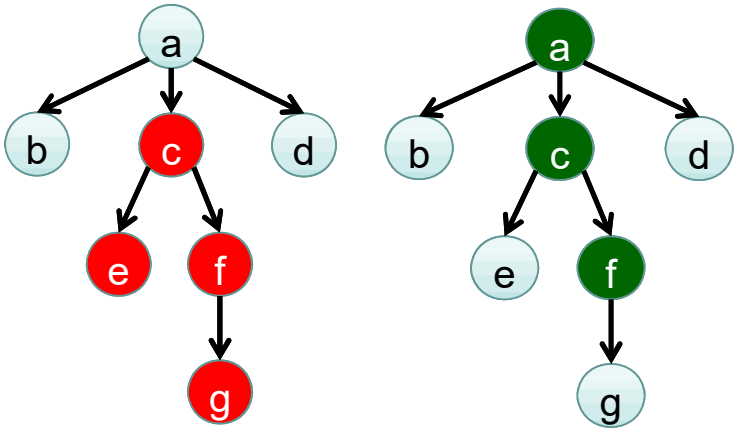
Definiciones (I)

- Un **Árbol** consiste en un nodo (**r**, denominado **nodo raiz**) y una **lista** o **conjunto** de subárboles (A_1, A_2, .. A_k).
- Si el orden de los subárboles importa, entonces forman una lista, y se denomina árbol ordenado (por defecto un árbol se supone que es ordenado). En caso contrario los subárboles forman un conjunto, y se denomina árbol no ordenado.
- Se definen como **nodos hijos** de **r** a los nodos raices de los subárboles A_1, A_2, .. A_k
- Si **b** es un nodo hijo de **a** entonces **a** es el **nodo padre** de **b**
- Un nodo puede tener cero o más hijos, y uno o níngun padre. El único nodo que no tiene padre es el nodo raíz del árbol.
- Un nodo sin hijos se denomina **nodo hoja o externo**. En caso contrario se denomina **nodo interno**.

Definiciones (II)

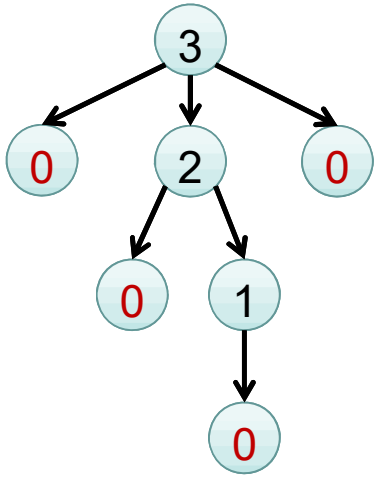
- Se define un **camino** en un arbol como cualquier secuencia de nodos del arbol, $n_1 \ldots n_p$, que cumpla que cada nodo es padre del siguiente en la secuencia (es decir, que n_i es el padre de n_{i+1}). La longitud del camino se define como el número de nodos de la secuencia menos uno ($p-1$).

- Los **descendientes** de un nodo (**c** en el diagrama) son aquellos nodos accesibles por un camino que comience en el nodo.
- Los **ascendientes** de un nodo (**f** en el diagrama) son los nodos del camino que va desde la raiz a él.



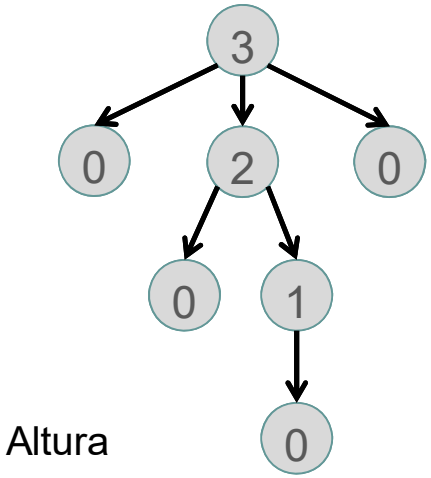
Altura

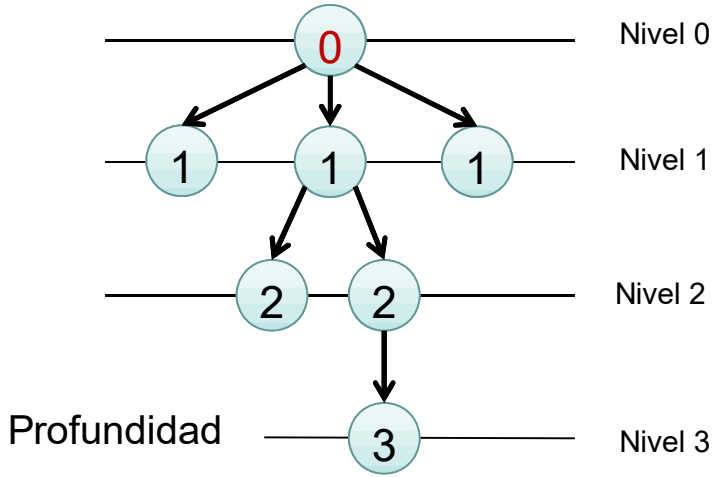
- Se define la **altura de un nodo** en un arbol como la longitud del camino más largo que comienza en el nodo y termina en una hoja.
  - La altura de un nodo hoja es 0
  - La altura de un nodo es igual a la mayor altura de sus hijos + 1
- La **altura de un árbol** se define como la altura de la raiz.
- La altura de un arbol determina la eficiencia de la mayoría de operaciones definidas sobre árboles.



Profundidad

- Se define la **profundidad de un nodo** en un arbol como la longitud del camino (único) que comienza en la raíz y termina en el nodo. También se denomina nivel.
  - La profundidad de la raiz es 0
  - La profundidad de un nodo es igual a la profundidad de su padre + 1



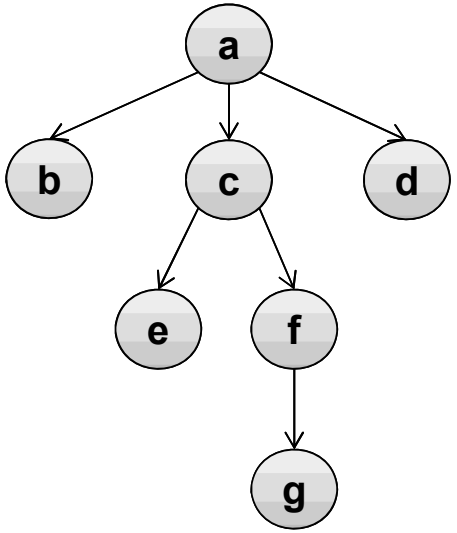


Recorrido de árboles

- **Preorden**: Se pasa por la raiz y luego se recorre en preorden cada uno de los subárboles. Recursivo.
- **Postorden**: Se recorre en postorden cada uno de los subárboles y luego se pasa por la raiz. Recursivo.
- **Inorden**: Se recorre en inorden el primer subárbol (si existe). Se pasa por la raíz y por último se recorre en inorden cada uno de los subárboles restantes. Tiene sentido fundamentalmente en árboles binarios. Recursivo.
- **Por Niveles**: Se etiquetan los nodos según su profundidad (nivel). Se recorren ordenados de menor a mayor nivel, a igualdad de nivel se recorren de izquierda a derecha.
  - No recursivo: Se introduce el raiz en una cola y se entra en un bucle en el que se extrae de la cola un nodo, se recorre su elemento y se insertan sus hijos en la cola.

Recorrido de árboles (II)

- **Preorden**: a,b,c,e,f,g,d
- **Postorden**: b,e,g,f,c,d,a
- **Inorden**: b,a,e,c,g,f,d
- **Por Niveles**: a,b,c,d,e,f,g

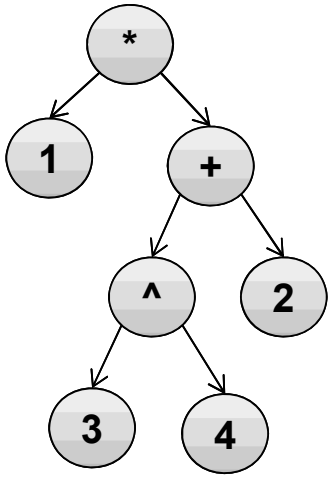


Parentizado sobre subárboles:

- **Preorden**: a (b) (c (e) (f (g))) (d)
- **Postorden**: (b) ((e) ((g) f) c) (d) a
- **Inorden**: (b) a ((e) c ((g) f)) (d)
- **Por Niveles**: (a) (b c d) (e f) (g)

Expresiones matemáticas

- **Preorden** →Notación prefija : * 1 + ^ 3 4 2
- **Postorden** → Notación postfija: 1 3 4 ^ 2 + *
- **Inorden** → Notación habitual: 1 * ((3 ^ 4) + 2)



Evaluación de expresiones

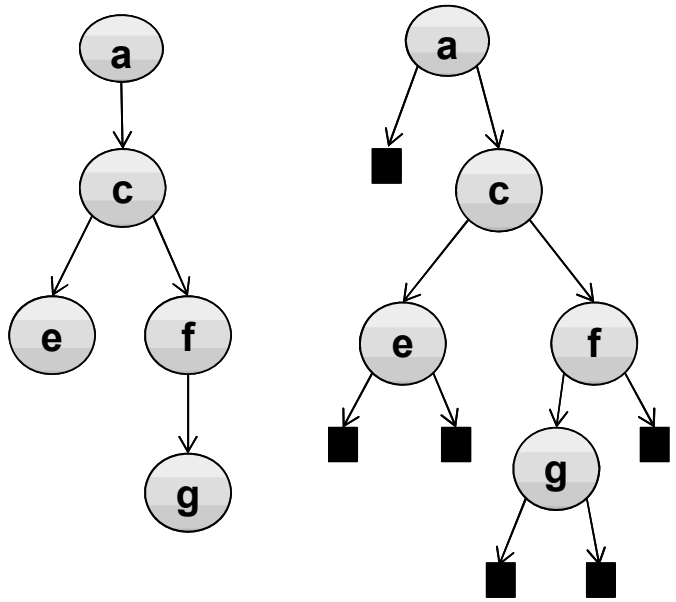
Se recorre el arbol en **postorden**:
Si es un operando, se inserta en **pila**
Si es un operador:
- Se extraen dos operandos
- Se aplica el operador
- Se inserta en pila el resultado

Al final, la pila debe contener un único valor, el resultado.

3. ÁRBOLES BINARIOS

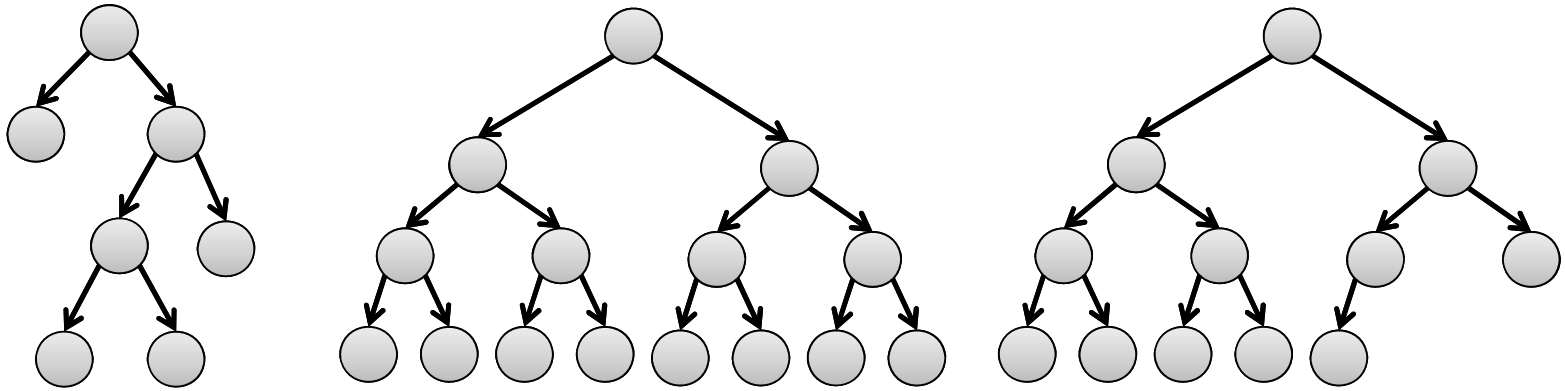
Árboles binarios

- **Árbol binario**: Es un árbol que o bien esta **vacío** (sin contenido) o bien consta de un nodo raiz con **dos** subárboles binarios, denominados **izquierdo** y **derecho**.
  - La existencia de árboles vacíos es una convención para que no exista ambigüedad al identificar el subarbol izquierdo y derecho. Se representa por un cuadrado.
  - La altura de un árbol vacío es -1
  - Cada nodo puede tener 0 hijos (subárbol izquierdo y derecho vacíos), 1 hijo (algún subárbol vacío) o 2 hijos.



Variantes de árboles binarios

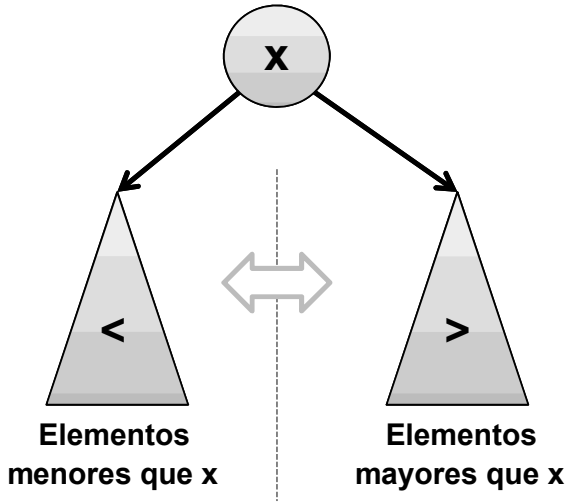
- **Árbol estricto**: Si un subárbol está vacío, el otro también. Cada nodo puede tener 0 ó 2 hijos.
- **Árbol lleno**: Árbol estricto donde en cada nodo la altura del subárbol izquierdo es igual a la del derecho, y ambos subárboles son árboles llenos.
- **Árbol completo**: Arbol lleno hasta el penúltimo nivel. En el último nivel los nodos están agrupados a la izquierda.



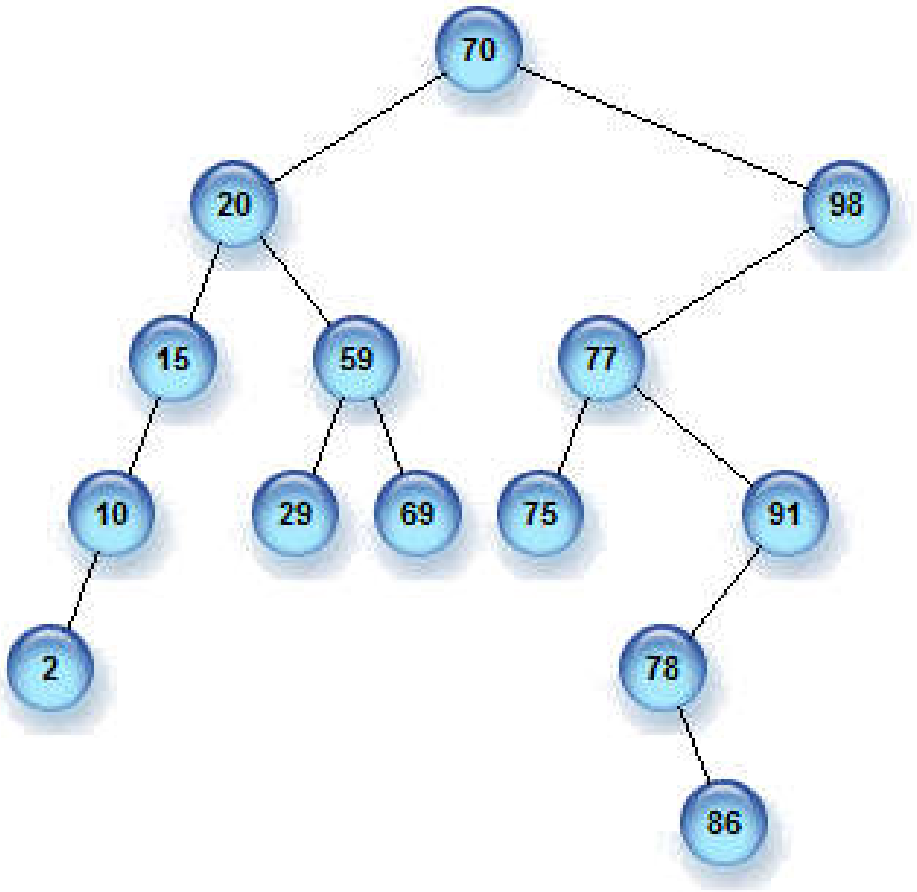
5. ÁRBOLES BINARIOS DE BÚSQUEDA

Árbol Binario de Búsqueda

- Un **árbol binario de búsqueda** (árbol BB) es un **árbol binario** cuyos nodos almacenan elementos comparables mediante $\leq$ y donde todo nodo cumple la **propiedad de ordenación**:
- **Propiedad de ordenación**: Todo nodo es **mayor** que los nodos de su subárbol **izquierdo**, y **menor** que los nodos de su subárbol **derecho**.



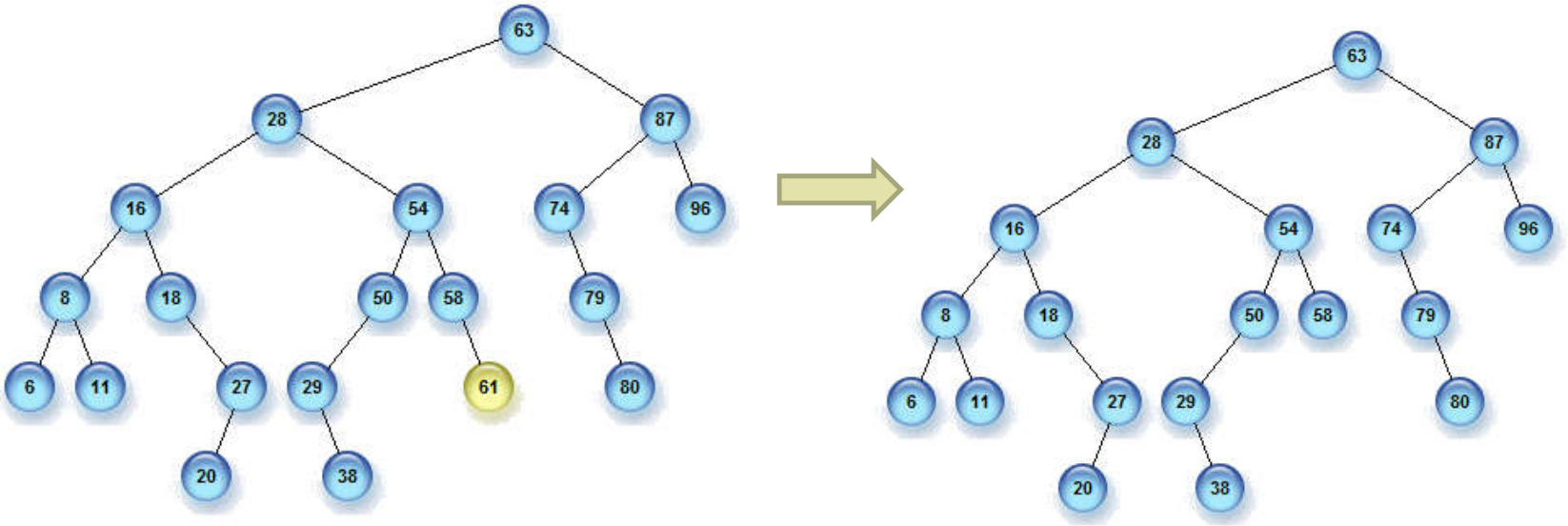
Ejemplo de árbol BB



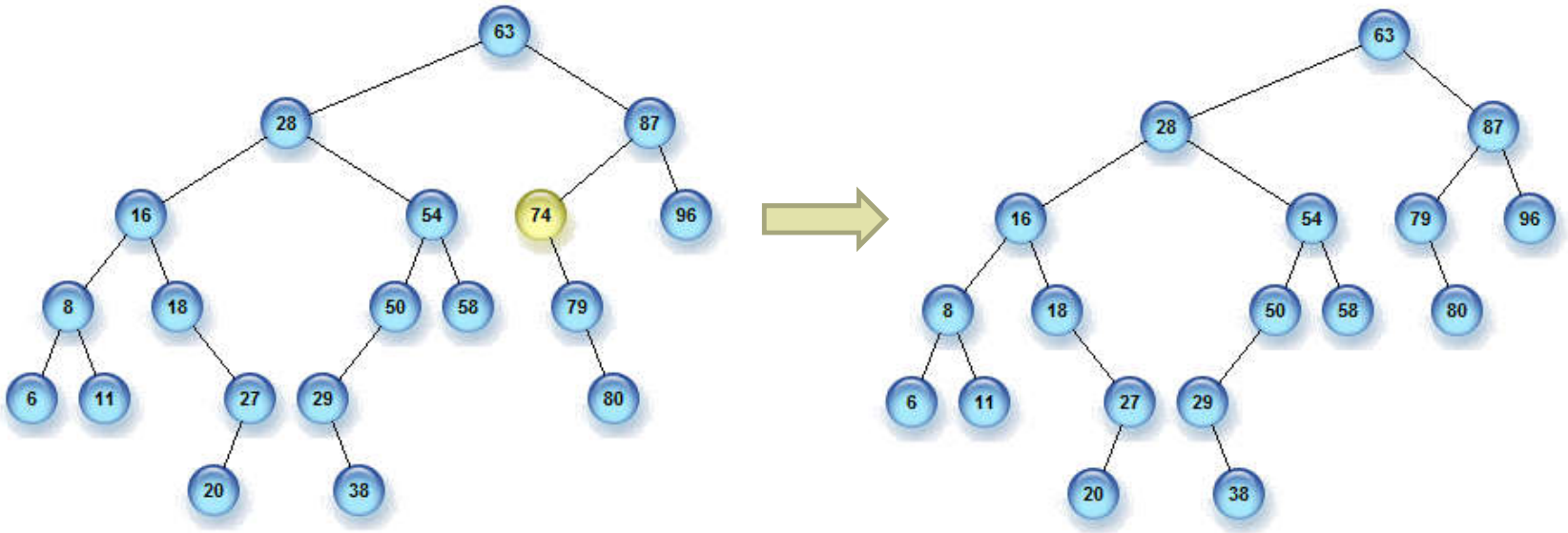
Propiedades y operaciones

- Un recorrido **inorden** por el árbol recorre los elementos en **orden** de menor a mayor.
- El elemento **mínimo** es el primer nodo sin hijo izquierdo en un descenso por hijos izquierdos desde la raiz.
- El elemento **máximo** es el primer nodo sin hijo derecho en un descenso por hijos derechos desde la raiz.
- Para **buscar** un elemento se parte de la raiz y se desciende escogiendo el subárbol izquierdo si el valor buscado es menor que el del nodo o el subárbol derecho si es mayor.
- Para **insertar** un elemento se busca en el árbol y se inserta como nodo hoja en el punto donde debería encontrarse.
- Para **borrar** un elemento, se adaptan los enlaces si tiene 0 o 1 hijo. Si tiene dos hijos se intercambia con el máximo de su subárbol izquierdo y se borra ese máximo.

Ejemplo de borrado – 0 hijos



Ejemplo de borrado – 1 hijos



Ejemplo de borrado – 2 hijos