

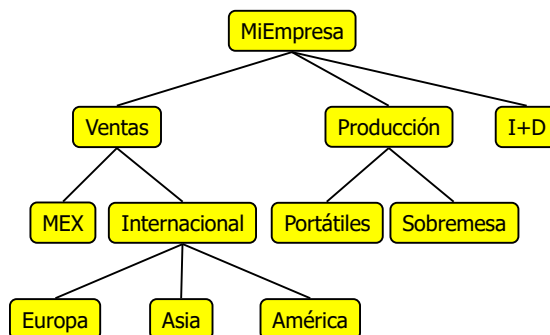
ÁRBOLES

Conceptos y
terminología

M.C. YALU GALICIA HDEZ. (FCC/BUAP) 1

CONCEPTO DE ÁRBOL

- Estructura jerárquica no lineal
- Relaciones padre-hijo entre nodos
- Ejemplos: sistema de archivos, estructura de un libro, diagrama modular, eliminatorias deportivas...



M.C. YALU GALICIA HDEZ. (FCC/BUAP) 2

CONCEPTO DE ÁRBOL

- Un árbol se caracteriza por estar formado por una serie de nodos conectados por aristas que verifican que:
 - Hay un único nodo raíz
 - Cada nodo, excepto la raíz, tiene un único padre
 - Hay un único camino (desde la raíz hasta cada nodo)

M.C. YALU GALICIA HDEZ. (FCC/BUAP) 3

CONCEPTO DE ÁRBOL

- Un árbol puede definirse de forma recursiva como:
 - Una colección de nodos que puede ser vacía, o que en su defecto consiste de un nodo raíz R y un número finito de estructuras tipo árbol T_1, \dots, T_k , llamados subárboles, los cuales son disjuntos y sus respectivos nodos raíz están conectados a R .
- Por tanto, un árbol es una estructura no secuencial.

M.C. YALU GALICIA HDEZ. (FCC/BUAP) 4

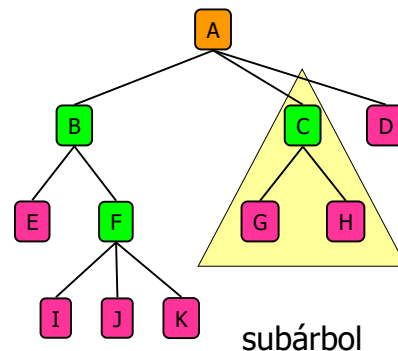
CONCEPTO DE ÁRBOL

- Un árbol es:
 - La estructura vacía, o
 - Un nodo de tipo árbol con un número finito de estructuras árbol disjuntas llamadas subárboles
- $$\langle \text{arbol} \rangle ::= \langle \text{nulo} \rangle \mid \langle \text{nodo} \rangle$$
- $$\langle \text{nodo} \rangle ::= \langle \text{info} \rangle \{ \langle \text{arbol} \rangle \}$$
- Puede estar ordenado o no

M.C. YALU GALICIA HDEZ. (FCC/BUAP) 5

TERMINOLOGÍA BÁSICA

- **Raíz:** único nodo sin padre
- **Nodo interno:** tiene al menos un hijo
- **Nodo hoja (externo):** no tiene hijos
- **Descendiente directo:** hijo
- **Descendientes:** hijos, nietos...
- **Subárbol:** árbol formado por un nodo y sus descendientes



M.C. YALU GALICIA HDEZ. (FCC/BUAP) 6

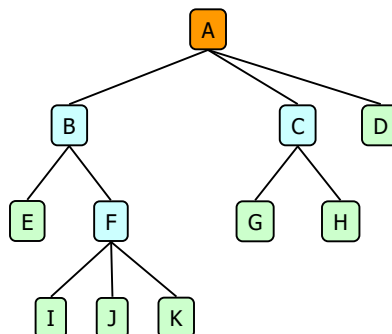
TERMINOLOGÍA BÁSICA

- **Grado de un nodo:** número de descendientes directos
- **Grado del árbol:** mayor grado de sus nodos
- **Árbol binario:** árbol de grado 2
 - Cada nodo tiene cuando más dos descendientes directos
- **Árbol multiamino** (o general):
 - Cada nodo puede tener n descendientes directos
- **Lista**
 - árbol degenerado de grado 1

M.C. YALU GALICIA HDEZ. (FCC/BUAP) 7

TERMINOLOGÍA BÁSICA

- **Profundidad de un nodo:** número de predecesores
- **Altura del árbol:** profundidad máxima de cualquier nodo



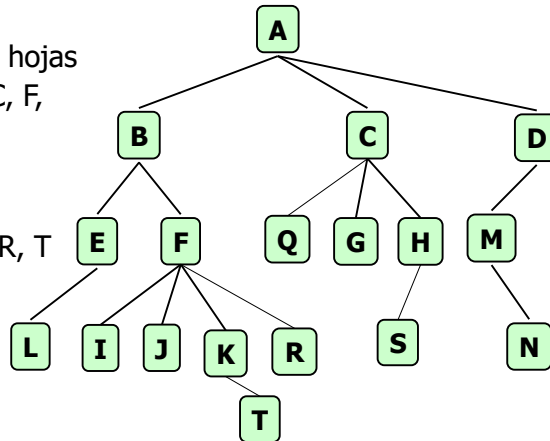
profundidad(A)=0
profundidad(H)=2
profundidad(K)=3
altura=3

M.C. YALU GALICIA HDEZ. (FCC/BUAP) 8

ACTIVIDAD INDIVIDUAL



- Para el árbol siguiente obtener:
 - No. de nodos internos y hojas
 - No. de descendientes (C, F, M)
 - Grado de A, C, F, D, T
 - Grado del árbol
 - Profundidad de A, F, Q, R, T
 - Altura del árbol



M.C. YALU GALICIA HDEZ. (FCC/BUAP)

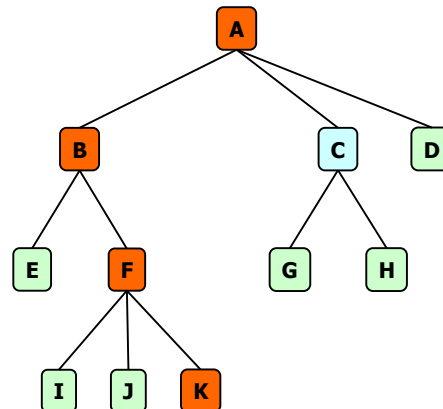
9

TERMINOLOGÍA BÁSICA

- **Camino:** existe un camino del nodo X al nodo Y, si existe una sucesión de nodos que permitan llegar desde X a Y.

$\text{camino}(A, K) = \{A, B, F, K\}$

$\text{camino}(C, K) = \{ \}$

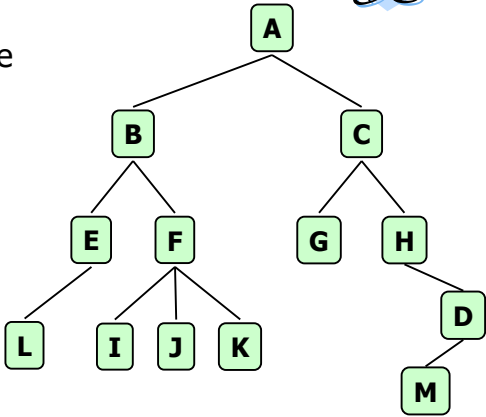


M.C. YALU GALICIA HDEZ. (FCC/BUAP) 10

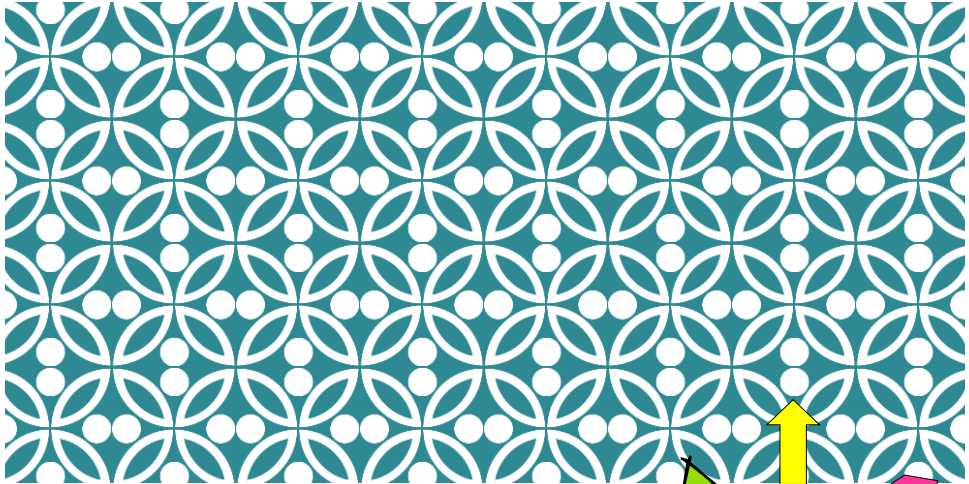
ACTIVIDAD INDIVIDUAL



- Para el árbol siguiente obtener el camino:
- C → M
- G → K
- A → I



M.C. YALU GALICIA HDEZ. (FCC/BUAP) 11

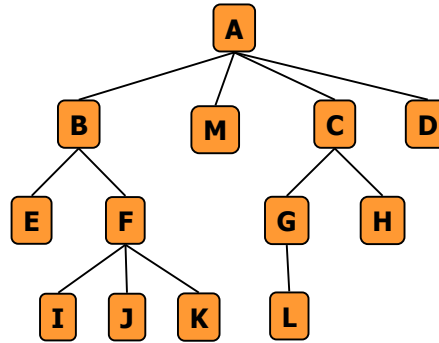


ÁRBOLES GENERALES O MULTICAMINOS



ÁRBOLES GENERALES O MULTICAMINO

- Árboles n-arios o Multicamino (NIST):
 - Árbol en el que cualquier nodo puede tener cualquier número de hijos
- ↓
- Árboles con grado ≥ 2

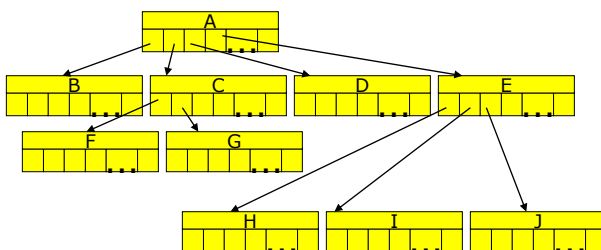


M.C. YALU GALICIA HDEZ. (FCC/BUAP)

13

ÁRBOLES GENERALES

- Implementación 1
 - Hijos como arreglos de referencias
 - Desaprovecha memoria si el número de hijos es muy variable
 - No puede usarse si el número de hijos es ilimitado

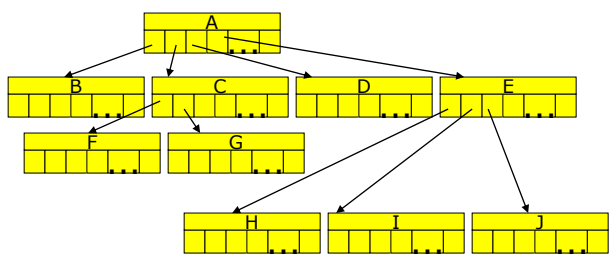


```
class Nodo{  
    Object info;  
    Nodo[] hijos;  
    .....  
}  
class ArbolMulticamino {  
    Nodo raiz;  
    .....  
}
```

M.C. YALU GALICIA HDEZ. (FCC/BUAP) 14

ÁRBOLES GENERALES

- Implementación 2
 - Hijos como un Vector/ArrayList
 - Lento en el acceso directo

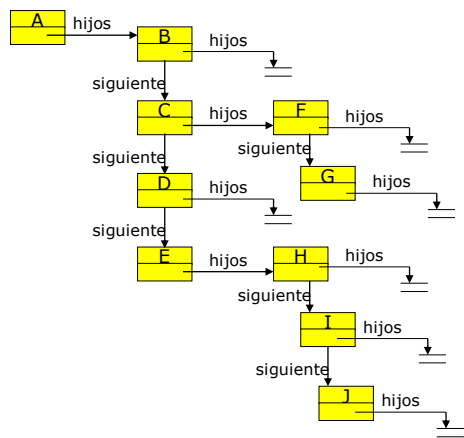


```
class Nodo{
  Object info;
  Vector hijos;
  ...
}
class ArbolMulticamino{
  Nodo raiz;
  ...
}
```

M.C. YALU GALICIA HDEZ. (FCC/BUAP) 15

ÁRBOLES GENERALES

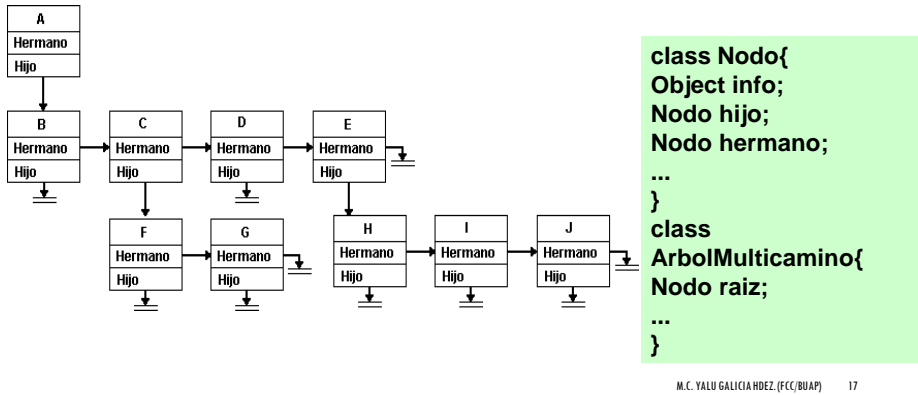
- Implementación 3
 - Hijos como una lista enlazada



```
class Nodo{
  Object info;
  ListaEnlazada hijos;
  ...
}
class ArbolMulticamino{
  Nodo raiz;
  ...
}
class ListaEnlazada{
  NodoLE inicio;
}
class NodoLE{
  Nodo nodoLista;
  NodoLE siguiente;
}
```


ÁRBOLES GENERALES

- Implementación 4 (*first child-next sibling binary tree*)
- Una referencia por cada tipo de relación: hijo/hermano



```
class Nodo{
  Object info;
  Nodo hijo;
  Nodo hermano;
  ...
}
class ArbolMulticamino{
  Nodo raiz;
  ...
}
```

M.C. YALU GALICIA HDEZ. (FCC/BUAP) 17

TDÁ ÁRBOL GENERAL: OPERACIONES

Creación de un árbol	crearArbol (nombreArbol)
Comprobación del edo.	arbolVacio(nombreArbol) → Booleano
Inserción de nodos	insertar(padre, valorInfo, posicion) insertar(padre, valorInfo)
Borrado de nodos	borrar(nombreArbol, valorInfo)
Búsqueda de un nodo	buscar(nombreArbol, dato) → informacion buscar(nombreArbol, información)→referenciaNodo
Recorrido del árbol	recorrer(nombreArbol,tipoRecorrido)
Acceso a los nodos	info(referenciaNodo) → Informacion primerHijo(referenciaNodo) → enlace hijos(referenciaNodo) → {enlace} eshoja(referenciaNodo) → Booleano
Modificación de los nodos	asignarInfo(referenciaNodo, valorInformacion) añadirHijo(referenciaNodo, valorInfo) quitarHijo(referenciaNodo, valorInfo)

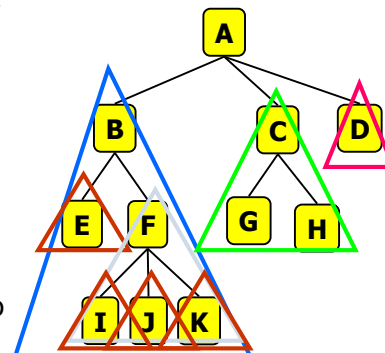
TAD ÁRBOL GENERAL: OPERACIONES

- Recorridos en árboles generales:
 - Pre-orden
 - Post-orden
 - Por niveles o primero en anchura (level-order traversal)

M.C. YALU GALICIA HDEZ. (FCC/BUAP) 19

ÁRBOL GENERAL: RECORRIDO PRE-ORDEN

- Recorrido en pre-orden:
 - Primero se visita o procesa la raíz del árbol
 - luego se visitan los subárboles que empiezan en sus hijos
 - se visita la raíz del subárbol
 - después se visitan los sub-subárboles que empiezan en sus hijos
 - Así sucesivamente hasta que no haya más subárboles
- ¿y el Algoritmo?



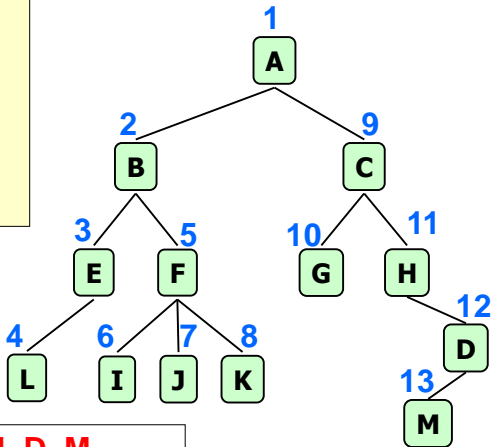
A, B, E, F, I, J, K, C, G, H, D

M.C. YALU GALICIA HDEZ. (FCC/BUAP)

20

OTRO EJEMPLO PREORDEN

```
preOrden(arbol)
  inicio
  procesar(arbol)
  Para cada hijo h de arbol
    preorden(h)
  fin_para
fin
```

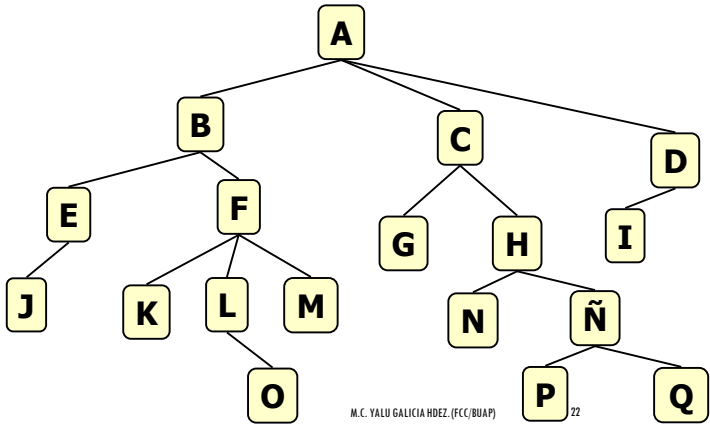


A, B, E, L, F, I, J, K, C, G, H, D, M

M.C. YALU GALICIA HDEZ. (FCC/BUAP) 21

ACTIVIDAD INDIVIDUAL

- Obtener el recorrido en **preOrden** del siguiente árbol general



M.C. YALU GALICIA HDEZ. (FCC/BUAP) 22

ÁRBOL GENERAL: RECORRIDO POST-ORDEN

■ Recorrido en **post-Orden**:

- los nodos se visitan después de haber visitado todos sus hijos

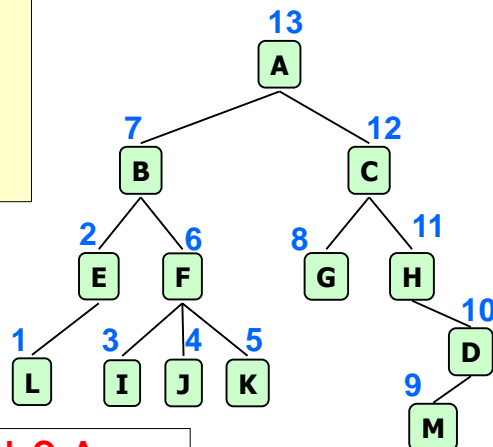
```
postOrden(arbol)  
inicio  
  Para cada hijo h de arbol  
    postOrden(h)  
  Fin_para  
  procesar(arbol)  
fin
```

M.C. YALU GALICIA HDEZ. (FCC/BUAP)

23

EJEMPLO POSTORDEN

```
postOrden(arbol)  
inicio  
  Para cada hijo h de arbol  
    postOrden(h)  
  Fin_para  
  procesar(arbol)  
fin
```



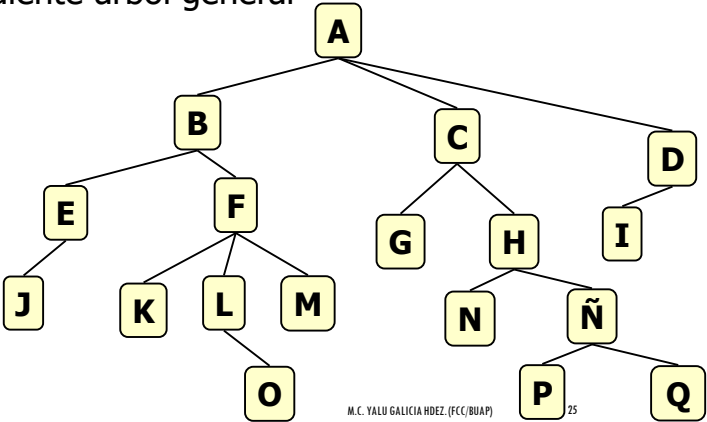
L, E, I, J, K, F, B, G, M, D, H, C, A

M.C. YALU GALICIA HDEZ. (FCC/BUAP) 24

ACTIVIDAD INDIVIDUAL

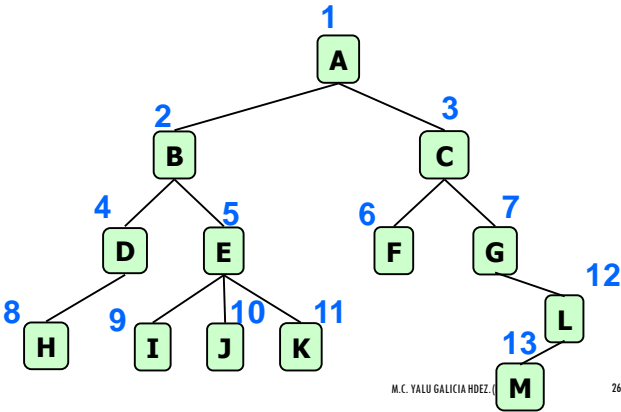


- Obtener el recorrido en **PostOrden** del siguiente árbol general



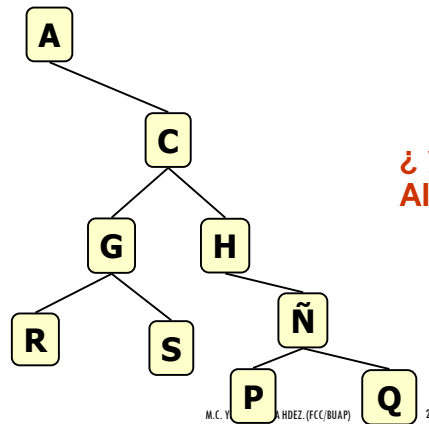
ÁRBOL GENERAL: RECORRIDO POR NIVELES

- Recorrido por niveles
 - se visita por niveles, de forma descendente y de izquierda a derecha



ACTIVIDAD INDIVIDUAL

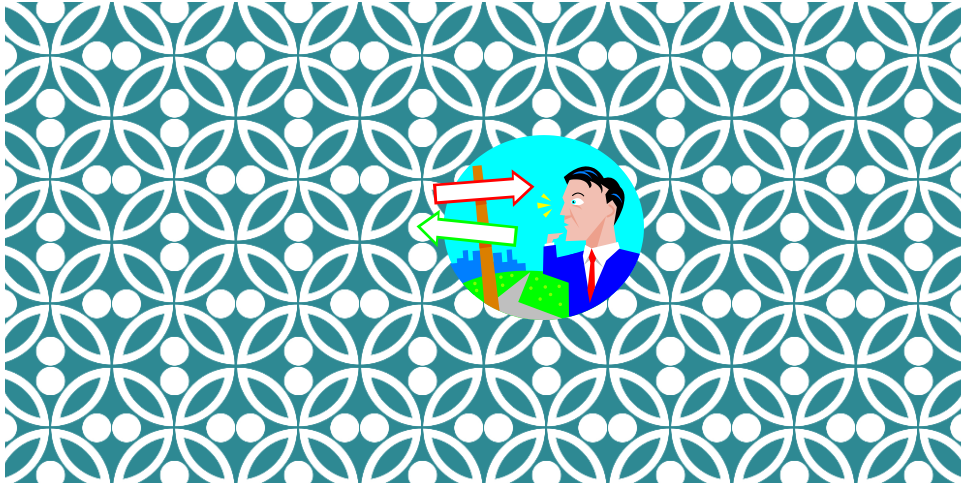
- Obtener el recorrido por niveles del siguiente árbol general



¿ y el
Algoritmo?

ÁRBOLES GENERALES: RECORRIDOS

- Ejemplo de aplicación
 - Pre-orden
 - Listar el contenido de un documento estructurado
 - Post-orden
 - Calcular el espacio en disco que ocupa un directorio

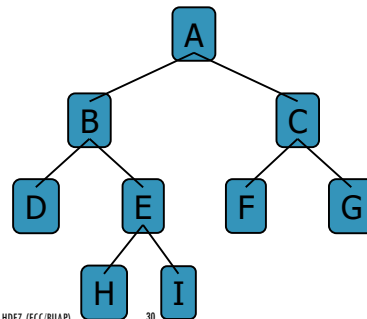


ÁRBOLES BINARIOS

M.C. YALU GALICIA HDEZ. (FCC/BUAP) 29

ÁRBOLES BINARIOS

- Es un árbol de grado 2
- Cada nodo tiene de 0 a 2 descendientes directos: el hijo izquierdo y el derecho
- Notación BNF (Backus-Naur Form)
 - `<arbol> ::= <<nulo>> | <nodo>`
 - `<nodo> ::= <info> <izq> <der>`
 - `<izq> ::= <arbol>`
 - `<der> ::= <arbol>`



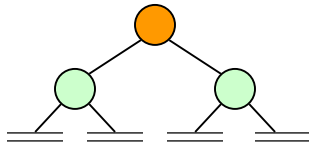
M.C. YALU GALICIA HDEZ. (FCC/BUAP)

30

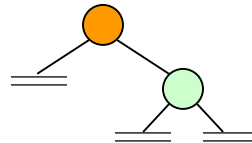
ÁRBOL BINARIO

- Aplicación: expresiones aritméticas, árboles de decisión, búsqueda (ABB)
- En algunos casos se exige que el árbol sea **completo** = todo nodo interno tiene dos descendientes.

Árbol binario completo



Árbol binario no completo



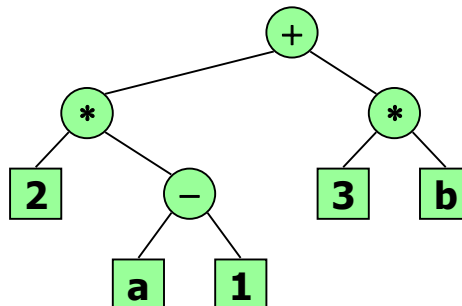
M.C. YALU GALICIA HDEZ. (FCC/BUAP)

31

ÁRBOL BINARIO

- Ejemplo: expresiones aritméticas
 - nodo interno: operadores
 - nodos hoja: operandos

$2*(a-1))+3*b$

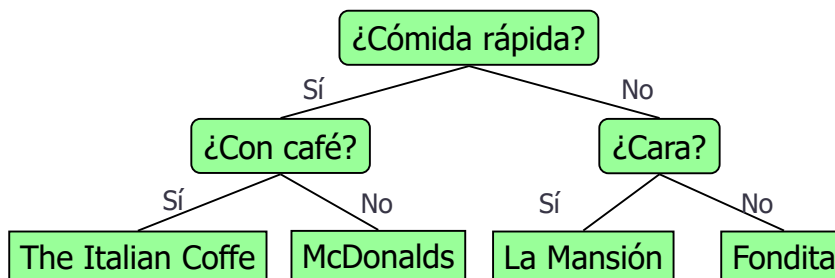


M.C. YALU GALICIA HDEZ. (FCC/BUAP)

32

ÁRBOL BINARIO

- Ejemplo de aplicación: árboles de decisión
 - nodo interno: preguntas con respuesta si/no
 - nodos hoja: decisiones
- ¿Dónde cenamos?



M.C. YALU GALICIA HDEZ. (FCC/BUAP)

33

TDÁ ÁRBOL BINARIO: OPERACIONES

Creación de un árbol	<code>crearArbol (nombreArbol)</code>
Comprobación del estado	<code>arbolVacio(nombreArbol) → Booleano</code>
Inserción de nodos	<code>insertar(padre, valorInfo, posicion)</code>
Borrado de nodos	<code>borrar(nombreArbol, valorInfo)</code>
Búsqueda de un nodo	<code>buscar(nombreArbol, dato) → informacion</code> <code>buscar(nombreArbol, informacion) → referenciaNodo</code>
Recorrido del árbol	<code>recorrer(nombreArbol, tipoRecorrido)</code>
Acceso a los nodos	<code>info(referenciaNodo) → Informacion</code> <code>izq(referenciaNodo) → enlace</code> <code>der(referenciaNodo) → enlace</code> <code>eshoja(referenciaNodo) → Booleano</code>
Modificación de los nodos	<code>asignarInfo(referenciaNodo, valorInformacion)</code> <code>asignarIzq(referenciaNodo, valorEnlace)</code> <code>asignarDer(referenciaNodo, valorEnlace)</code>

TAD ÁRBOL BINARIO: RECORRIDOS

- Hay tres tipos de recorrido en un árbol binario
 - in-Orden
 - pre-Orden
 - post-Orden

M.C. YALU GALICIA HDEZ. (FCC/BUAP) 35

ÁRBOL BINARIO: RECORRIDOS

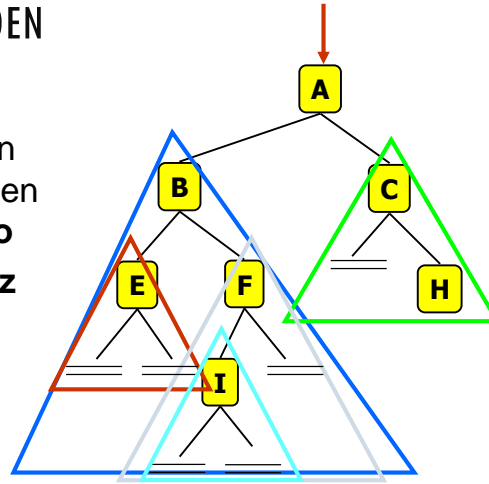
- Recorrido IN-ORDEN
 - Iniciando en la raíz, primero se efectúa un recorrido en InOrden en su subárbol izquierdo, luego se visita la raíz, y luego se visita el subárbol derecho también en InOrden
 - Es decir, cada nodo se visita tras visitar su subárbol izquierdo y antes de visitar el derecho

(izq, raíz, der)

M.C. YALU GALICIA HDEZ. (FCC/BUAP) 36

ABB: RECORRIDO IN-ORDEN

- Iniciando en la raíz
 - primero se efectúa un recorrido en InOrden en su subárbol **izquierdo**
 - luego se **visita** la **raíz**
 - y luego se visita el subárbol **derecho** también en InOrden
- **Clave: IRD**



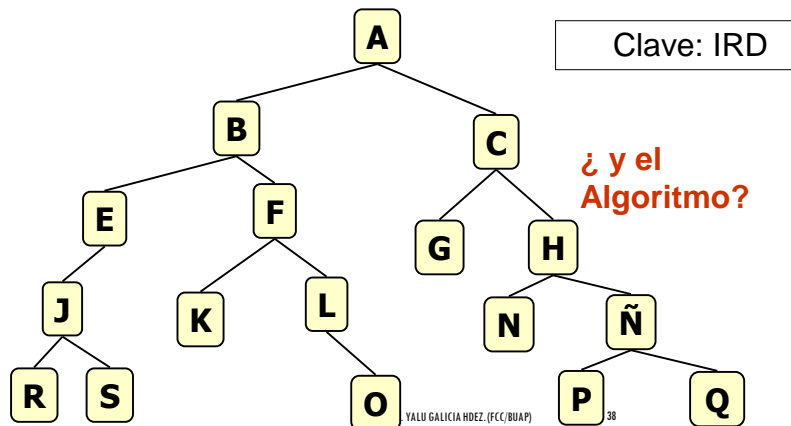
E, B, I, F, A, C, H

M.C. YALU GALICIA HDEZ. (FCC/BUAP)

37

ACTIVIDAD INDIVIDUAL

Obtener el recorrido **in-Orden** del ABB siguiente



Clave: IRD

¿ y el
Algoritmo?

M.C. YALU GALICIA HDEZ. (FCC/BUAP)

38

ÁRBOL BINARIO: RECORRIDOS

■ Recorrido PRE-ORDEN

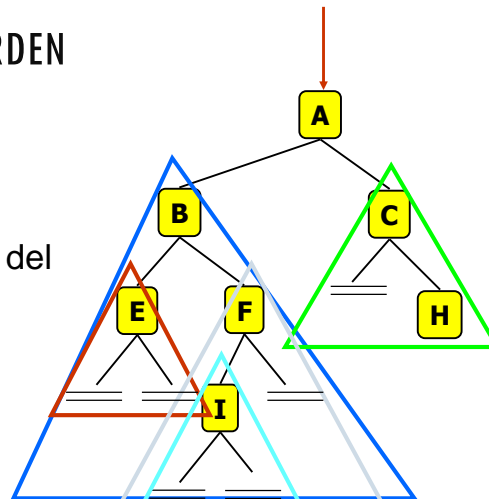
- ❑ Iniciando en la raíz, primero se visita ésta y luego se hace un recorrido en PreOrden del subárbol Izquierdo. Luego en el subárbol derecho, también en PreOrden
- ❑ Es decir, primero se visita cada nodo, luego su subárbol izquierdo y finalmente el derecho

(raiz, izq, der)

M.C. YALU GALICIA HDEZ. (FCC/BUAP) 39

ABB: RECORRIDO PRE-ORDEN

- Iniciando en la raíz, primero se visita ésta.
- luego se hace un recorrido en PreOrden del subárbol Izquierdo.
- Luego en el subárbol derecho, también en PreOrden
- **Clave: RID**



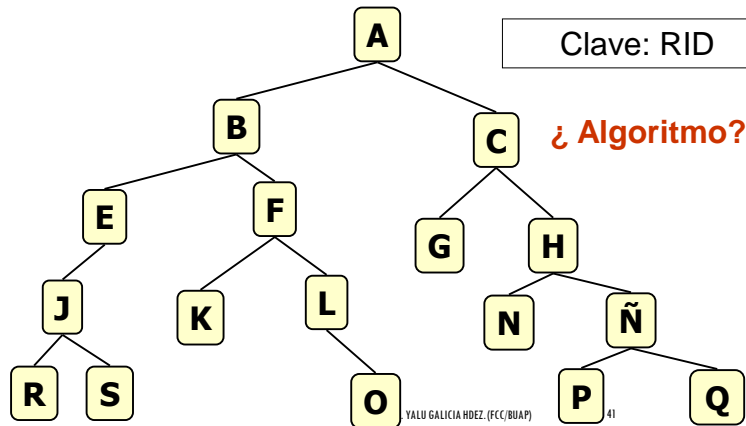
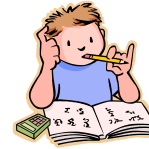
A, B, E, F, I, C, H

M.C. YALU GALICIA HDEZ. (FCC/BUAP)

40

ACTIVIDAD INDIVIDUAL

- Obtener el recorrido **pre-Orden** del ABB siguiente



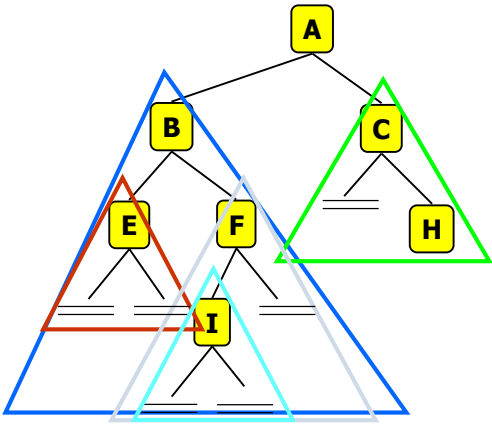
ÁRBOL BINARIO: RECORRIDOS

- Recorrido POST-ORDEN
 - Iniciando en la raíz, primero se visita en PostOrden el subárbol izquierdo, luego el subárbol derecho, también en PostOrden, y por último se visita la raíz
 - cada nodo se visita después de visitar su subárbol izquierdo y después de visitar el derecho

(izq, der, raiz)

ABB: RECORRIDO POST-ORDEN

- Iniciando en la raíz
 - primero se visita en PostOrden el subárbol izquierdo
 - luego el subárbol derecho, también en PostOrden
 - y por último se visita la raíz



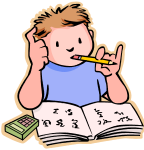
E, I, F, B, H, C, A

M.C. YALU GALICIA HDEZ. (FCC/BUAP)

43

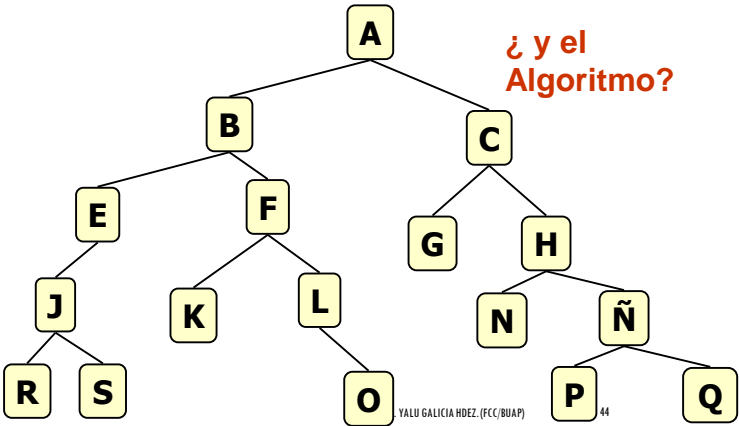
ACTIVIDAD INDIVIDUAL

- Obtener el recorrido **post-Orden** del ABB siguiente



Clave: IDR

¿ y el Algoritmo?

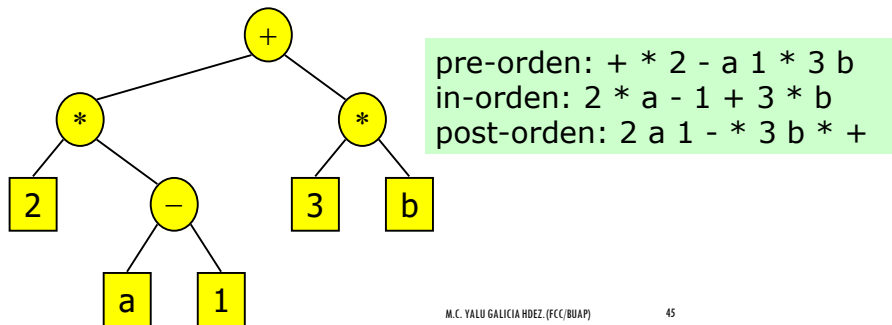


M.C. YALU GALICIA HDEZ. (FCC/BUAP)

44

AB: EJEMPLOS DE RECORRIDO

- Recorridos en las expresiones aritméticas
 - pre-orden: notación prefija (polaca)
 - in-orden: notación normal – infija (sin paréntesis)
 - post-orden: notación polaca inversa

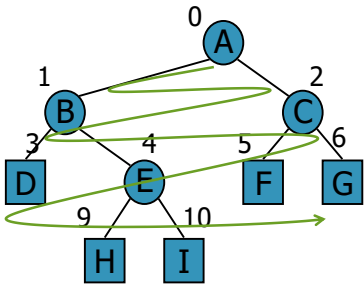


AB: APLICACIÓN DE RECORRIDOS

- Dibujar un árbol binario ***in-orden***
 - $x(v)$ = rango en in-orden de v
 - $y(v)$ = profundidad de v
- Imprimir una expresión aritmética ***in-orden***
- Evaluar una expresión aritmética ***post-orden***
- Imprimir el índice de un documento ***pre-orden***
- Calcular espacio total ocupado por una carpeta en sistema de ficheros ***post-orden***

AB: ESTRUCTURA ESTÁTICA

- Función **p** de numeración por niveles:
 - Si *v* es raíz de *T*, $p(v) = 0$
 - Si *v* es hijo izquierdo de *u*, $p(v) = 2 \cdot p(u) + 1$
 - Si *v* es hijo derecho de *u*, $p(v) = 2 \cdot p(u) + 2$
- La función *p* representa un polinomio de direccionamiento
- Representar el árbol con un arreglo donde cada nodo *v* se almacena en la posición *p(v)* del arreglo



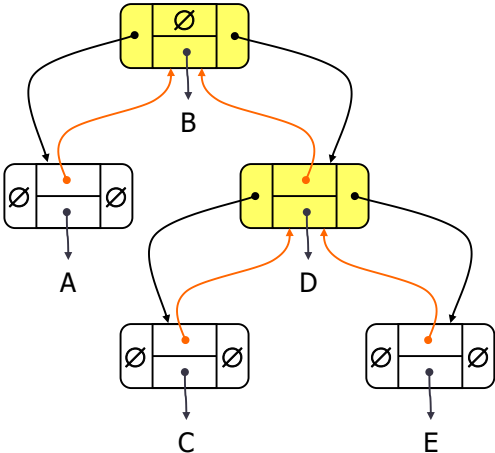
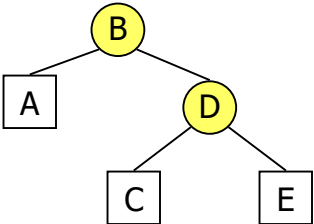
A	B	C	D	E	F	G			H	I		
0	1	2	3	4	5	6	7	8	9	10	11	12

- ¿En que posición irían los hijos de G?
- ¿En que posición los hijos de I?

(C/BUAP) 47

AB: ESTRUCTURA DINÁMICA

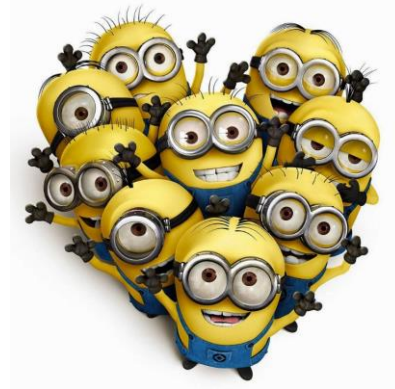
- Cada nodo contiene:
 - Elemento
 - Nodo hijo izquierdo
 - Nodo hijo derecho
 - [Nodo padre] Opcional
- Árbol = ref. al nodo raíz



M.C. YALU GALICIA HDEZ. (FCC/BUAP) 48

ACTIVIDAD COLABORATIVA

- Formen **equipos base** de 8 integrantes
- Cada integrante numérese del 1 al 8



YALÚ GALICIA HDEZ. (FCC-BUAP) 49

ACTIVIDAD COLABORATIVA

- El número que te tocó corresponde a uno de los siguientes problemas: Dado un árbol binario, obtener
 1. número de nodos hoja
 2. altura del árbol
 3. Número de nodos con solo 1 hijo
 4. Determinar si el árbol es completo
 5. Obtener en número de ocurrencias de un nodo x
 6. Profundidad de un nodo dado
 7. Número de nodos internos
 8. Número de nodos con exactamente dos hijos

YALÚ GALICIA HDEZ. (FCC-BUAP) 50

ACTIVIDAD COLABORATIVA

- Formar equipos de especialistas.
 - Cada integrante localice a los integrantes de otros equipos que les tocó el mismo número y formen un equipo de especialistas.
 - Tienen 15 mins, para que en equipo de especialistas resuelvan y entiendan el problema que les tocó.
 - Escriban en su libreta el algoritmo (solución al problema)
 - Cada uno lo explicará a su equipo base. También incluyan un ejemplo y prueba de escritorio

YALÚ GALICIA HDEZ (FCC-BUAP) 51

ACTIVIDAD COLABORATIVA

- Cada especialista regresa a su equipo base
 - En turnos cada integrante del equipo, explique el algoritmo del problema de su especialidad
 - Asegúrense que todos los integrantes de su equipo entienden su explicación.
 - Tomen notas, hagan preguntas...
 - Comenten en equipo cualquier duda o discrepancia que encuentren
- Prepárense para que cualquier integrante del equipo pueda explicar cualquier problema al resto del grupo

YALÚ GALICIA HDEZ (FCC-BUAP) 52

QUÉ HEMOS APRENDIDO?

