Tema 5. Árboles
Árboles Generales y Binarios

Estructura de Datos y Algoritmos (EDA)

Material elaborado por los profesores de la asignatura

Índice

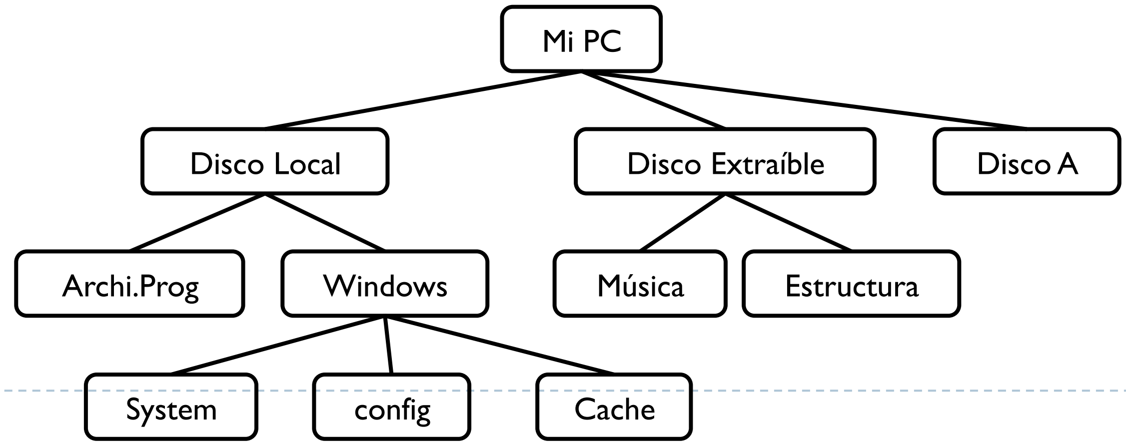
- **Conceptos básicos**
- TAD Árboles generales
- TAD Árboles binarios
- TAD Árboles binarios de búsqueda
- Equilibrado de árboles.

¿Qué es un árbol?

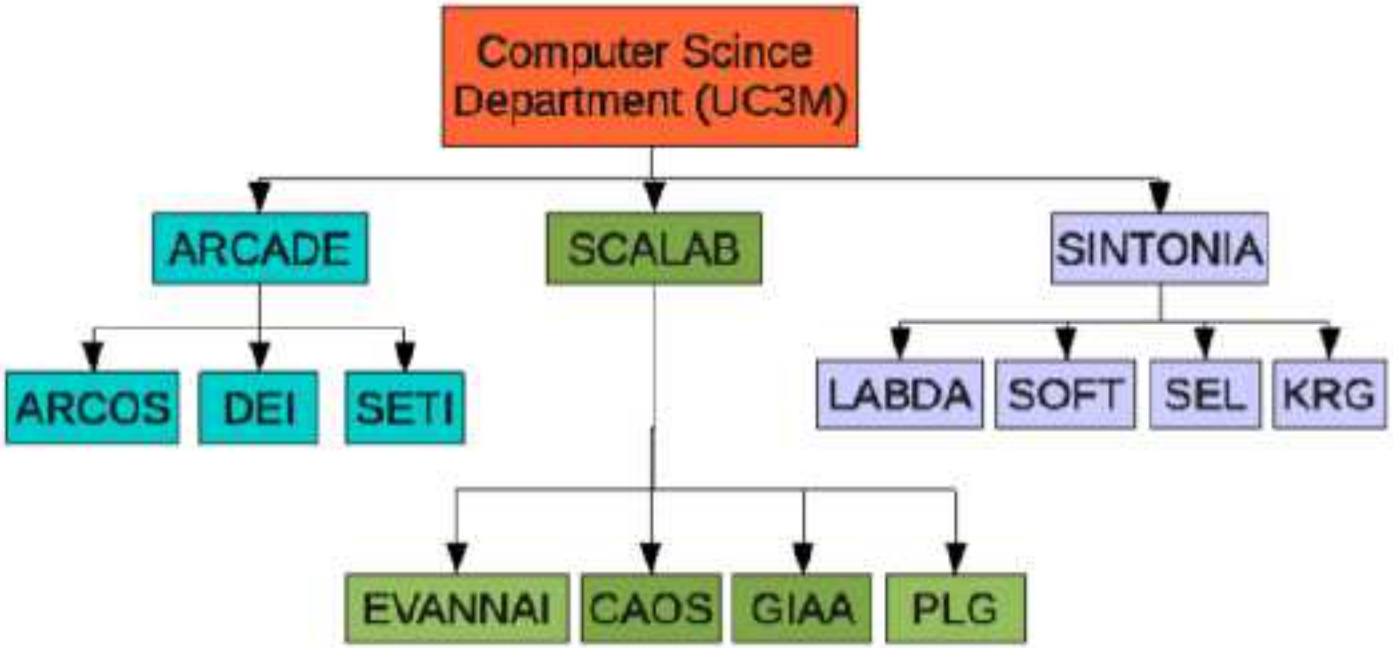
- Un árbol es un TAD que almacena elementos que tienen una relación jerárquica entre ellos (estructura jerárquica no lineal). El acceso a los elemento suele ser más rápido que en una estructura lineal.

- Relaciones padre-hijo entre nodos

- Ejemplos: **sistema de ficheros**, estructura de un libro, diagrama modular, bases de datos, interfaces gráficos, web sites, árbol genealógicos, organigramas, etc.



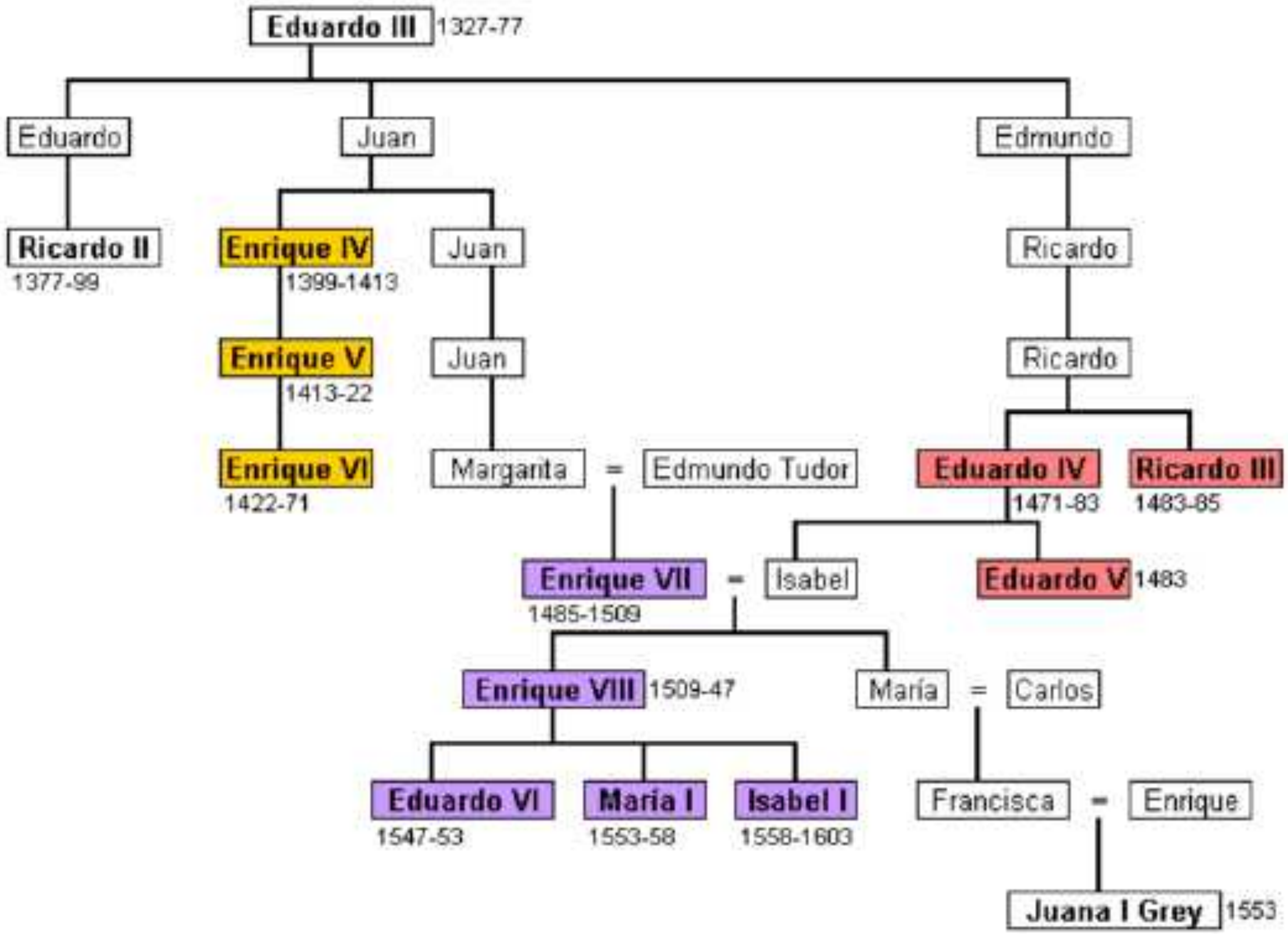
Ejemplo: organigrama de una organización



http://www.inf.uc3m.es/es/investigacion

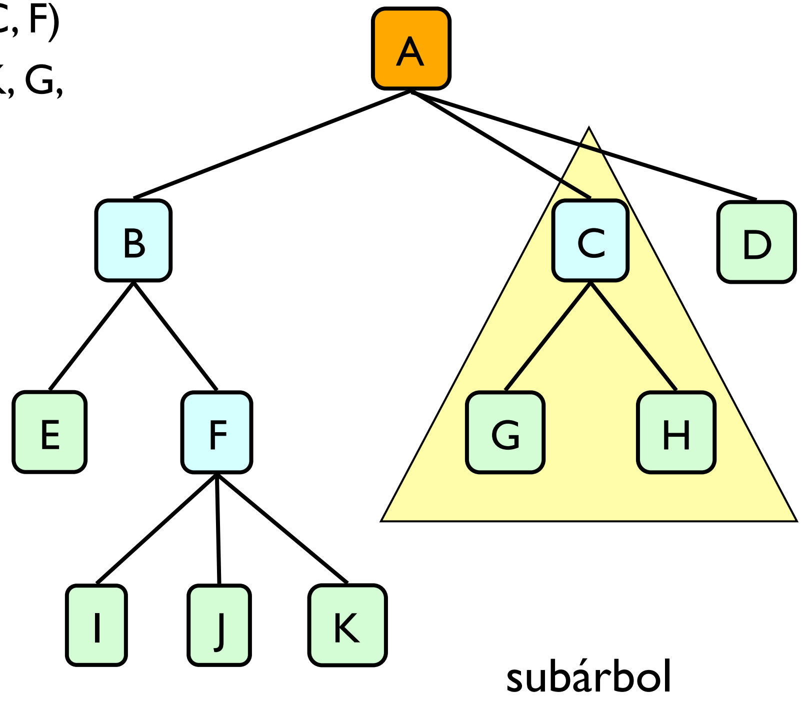
Ejemplos: árbol genealógico

FAMILIA TUDOR:



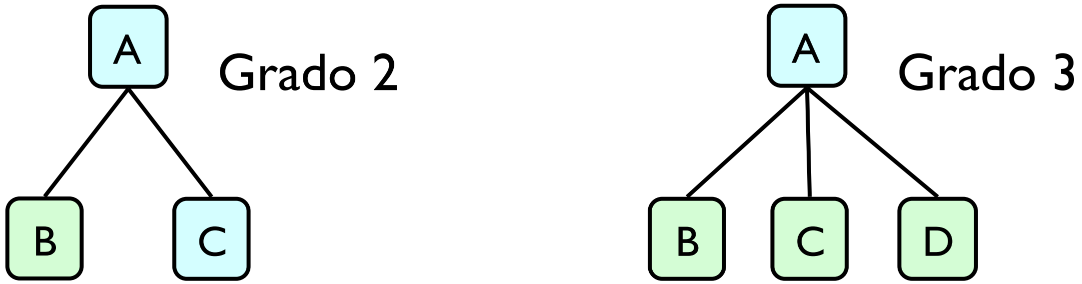
Conceptos básicos

- **Raíz**: único nodo sin padre (A)
- **Nodo interno**: tiene al menos un hijo (A, B, C, F)
- **Nodo hoja (externo)**: no tiene hijos (E, I, J, K, G, H, D)
- **Subárbol**: árbol formado por un nodo y sus descendientes
- **Ancestros y descendiente directos.**
- **Ancestros y descendientes.**



Conceptos Básicos

- **Grado de un nodo**: número de descendientes directos



- **Grado del árbol**: mayor grado de sus nodos

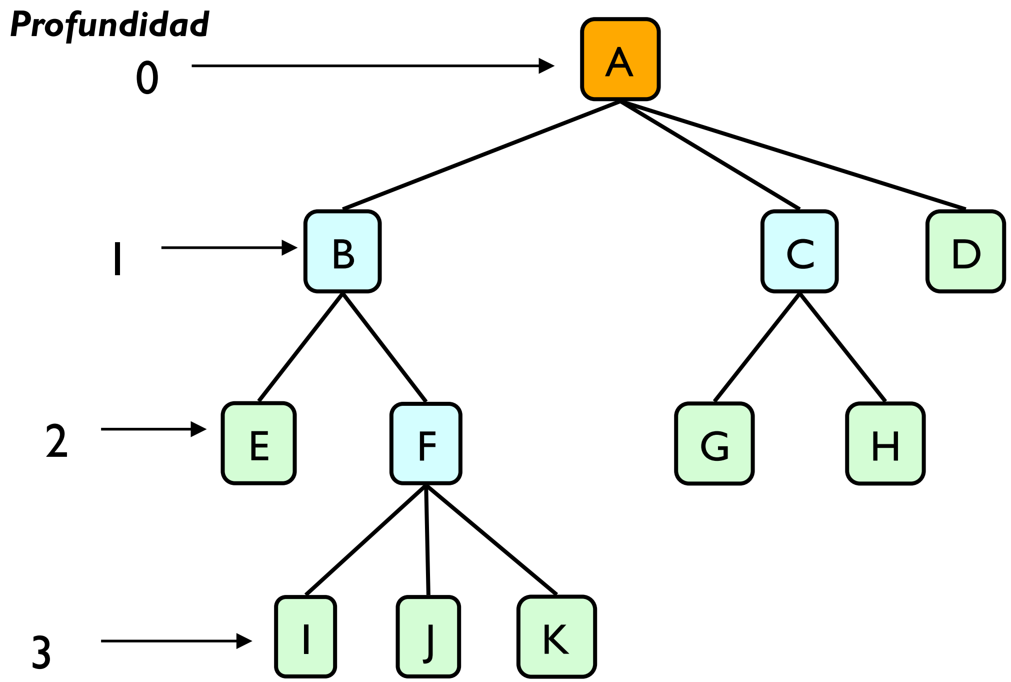
- **Ejemplos:**
  - **Árbol binario**: árbol de grado 2
    - Cada nodo tiene como mucho dos descendientes directos
  - **Lista:** árbol degenerado de grado 1

Cónceptos básicos

- **Profundidad de un nodo:** número de predecesores
- **Altura del árbol**: longitud de la rama más larga más uno.

profundidad(A)=0
profundidad(H)=2

Altura de árbol = 4

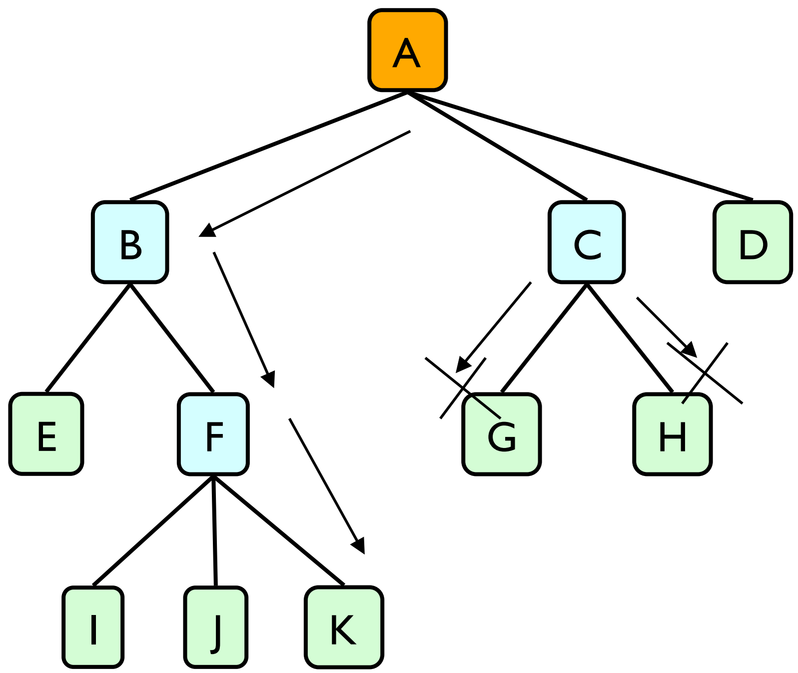


Conceptos básicos

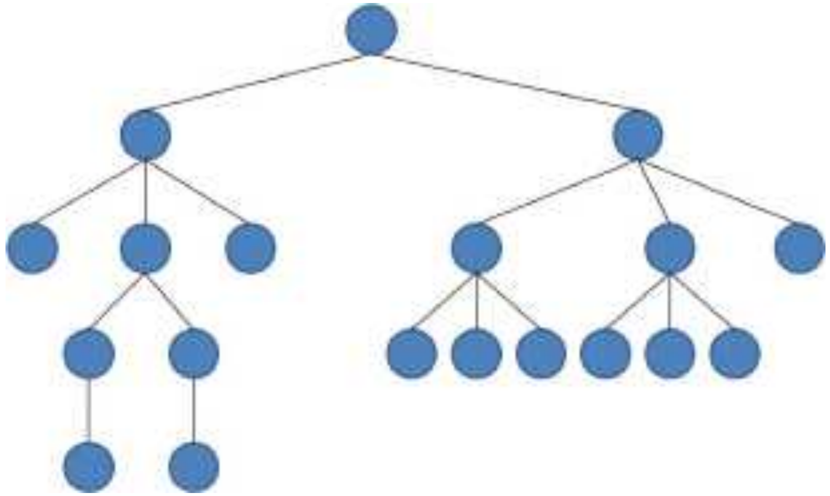
- **Camino:** existe un camino del nodo X al nodo Y, si existe una sucesión de nodos que permitan llegar desde X a Y. La sucesión debe tener un único sentido: ascendiente o descendiente.

camino(A,K)={A,B,F,K}

camino(C,K)={}



Árboles: Ejercicio 1



1. Explica los valores de las principales características del árbol mostrado en la figura.
   - ¿grado del árbol?
   - ¿altura del árbol?
   - ¿número nodos del árbol?
   - ¿número de hojas?
   - ¿número de nodos internos?

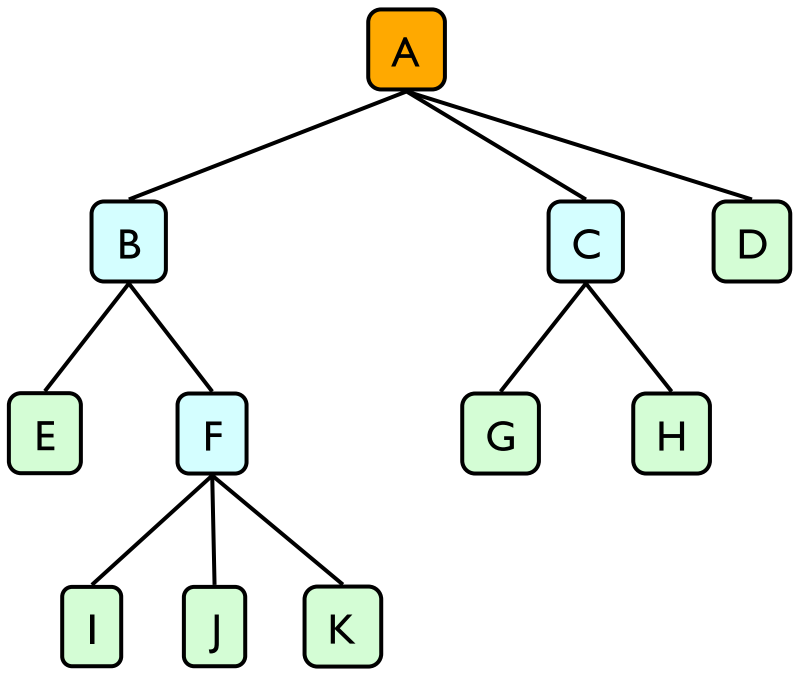
2. Dadas las siguientes propiedades de un árbol, proporcione un dibujo que satisfaga las mismas:
   - Grado del árbol: 3
   - Nº de Nodos: 14
   - Altura del árbol: 4
   - Nº nodos con profundidad 2: 6

Índice

- Conceptos básicos
- **TAD Árboles generales**
  - **Recorridos: post-orden, pre-orden y por niveles**
- TAD Árboles binarios
- TAD Árboles binarios de búsqueda
- TAD Árboles B

Árboles generales

- También llamados n-arios o multicamino
- Árboles con grado mayor a 2



Árboles generales: Aplicación

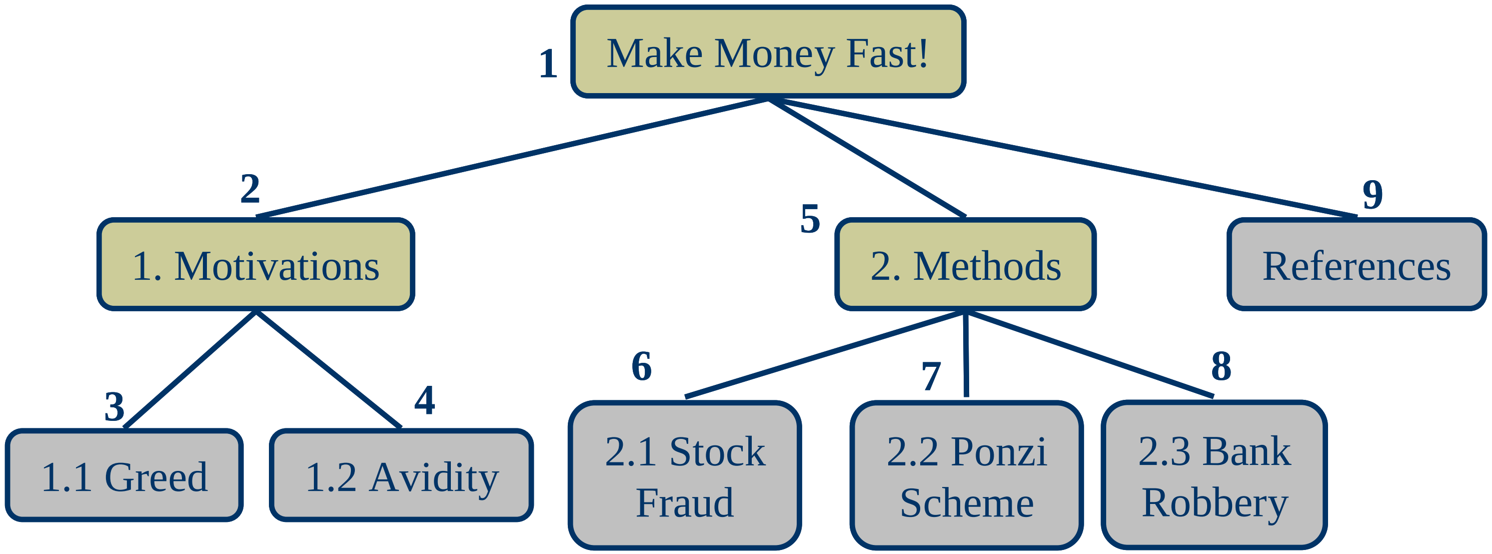
- Es la estructura utilizada para representar organizaciones jerárquicas donde cada elemento tiene un número variable de hijos.

- Ejemplos:
  - Representación de un sistema de ficheros, en el que cada directorio está enlazado con sus descendientes, ficheros o subdirectorios.
  - Representación de un árbol genealógico en el que cada persona se enlaza con sus descendientes

Árboles generales: Recorridos

Recorrido en pre-orden:

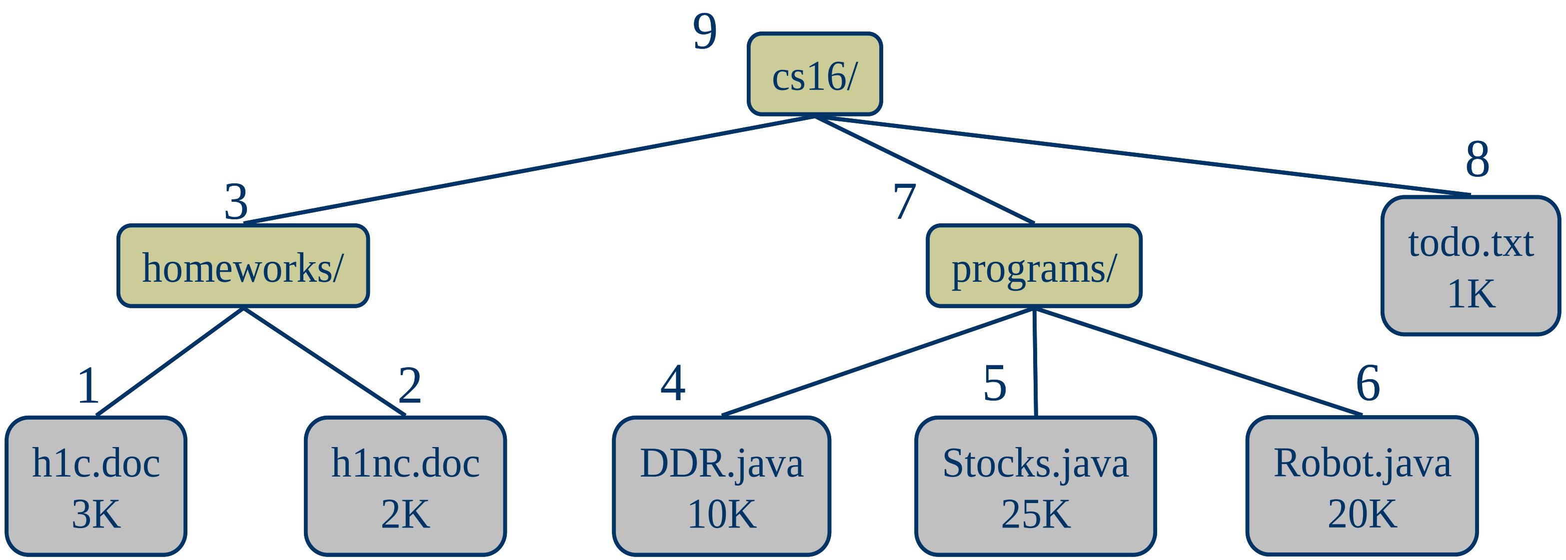
- se visita primero la raíz del árbol y luego se visitan recursivamente sus sub-árboles de izquierda a derecha (también en recorrido pre-orden)
- Ej, listar el contenido de un documento estructurado.



Árboles generales: Recorridos

Recorrido en post-orden:

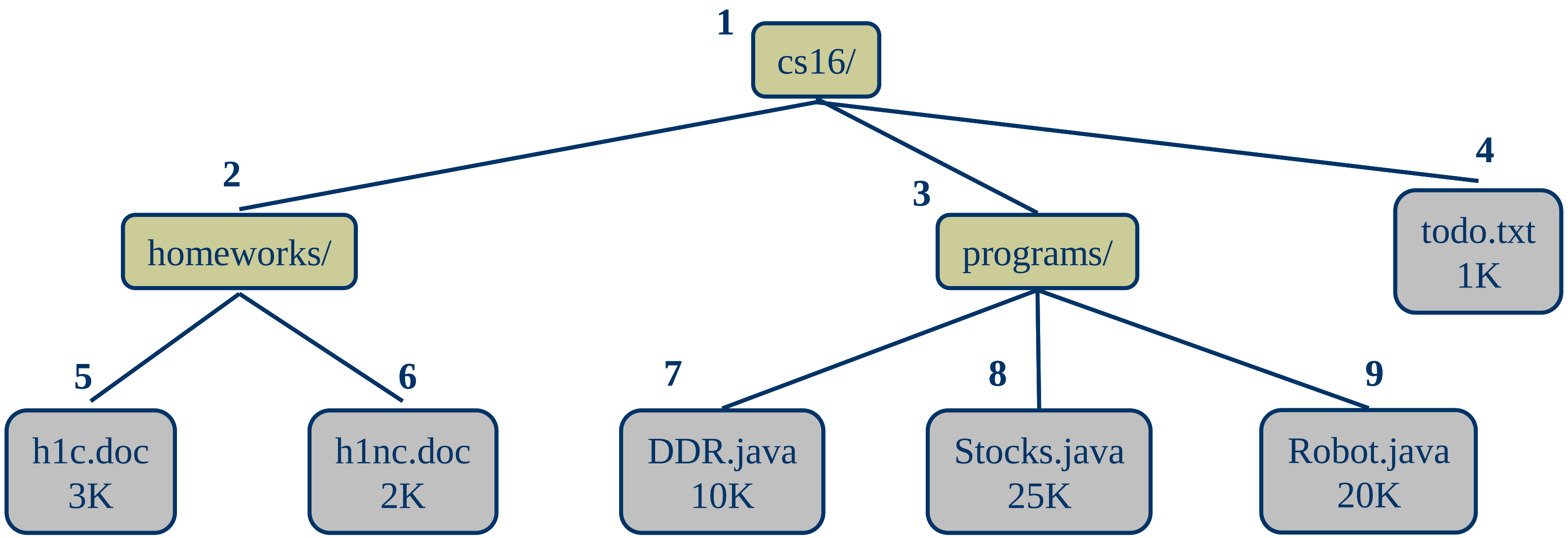
- los nodos se visitan después de haber visitado a sus hijos. Es decir, primero se recorren su sub-árboles (en recorrido post-orden) y por último se visita su raíz.
- Ej: Es útil para calcular el espacio en disco que ocupa un directorio.



Árboles generales: Recorridos

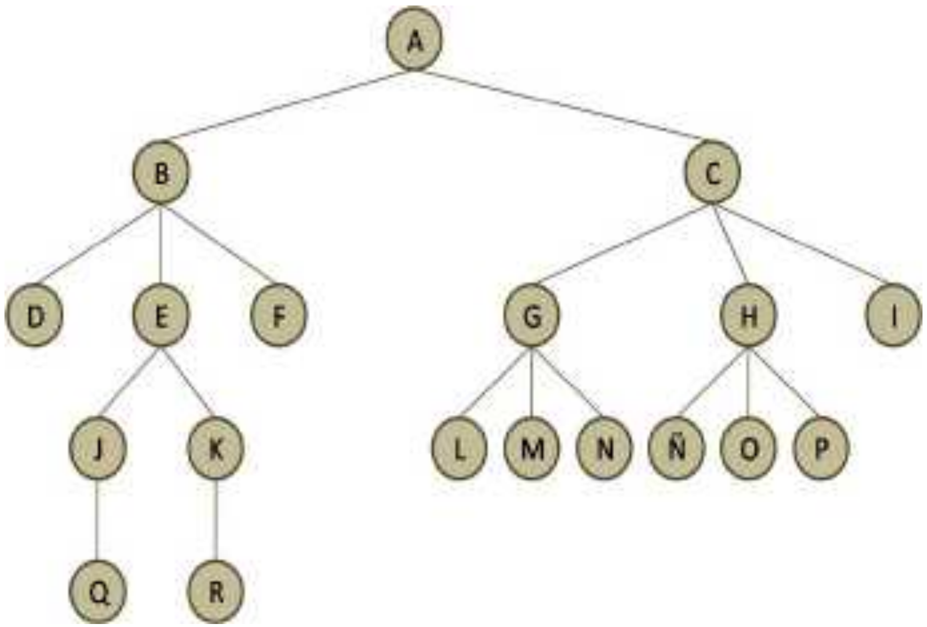
Recorridos por niveles:

- se visita por profundidad. Es decir, vamos visitando los nodos del mismo nivel de forma descendente y de izquierda a derecha

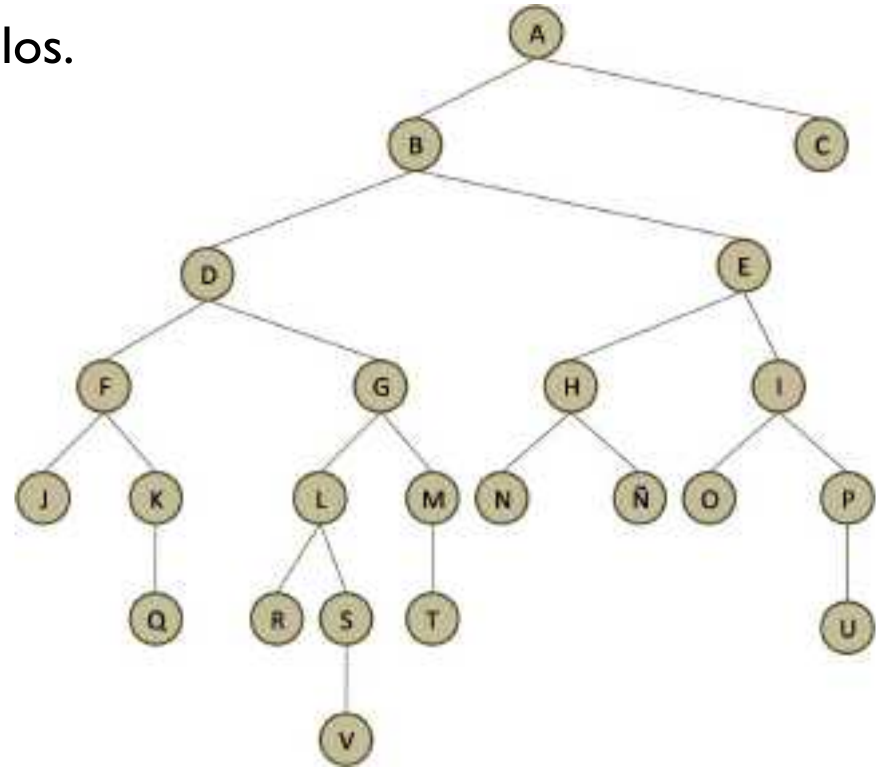


Árboles: Ejercicio 2

Dados los siguientes árboles, escriba los recorridos pre-orden y post-orden y por niveles de cada uno de ellos.



A



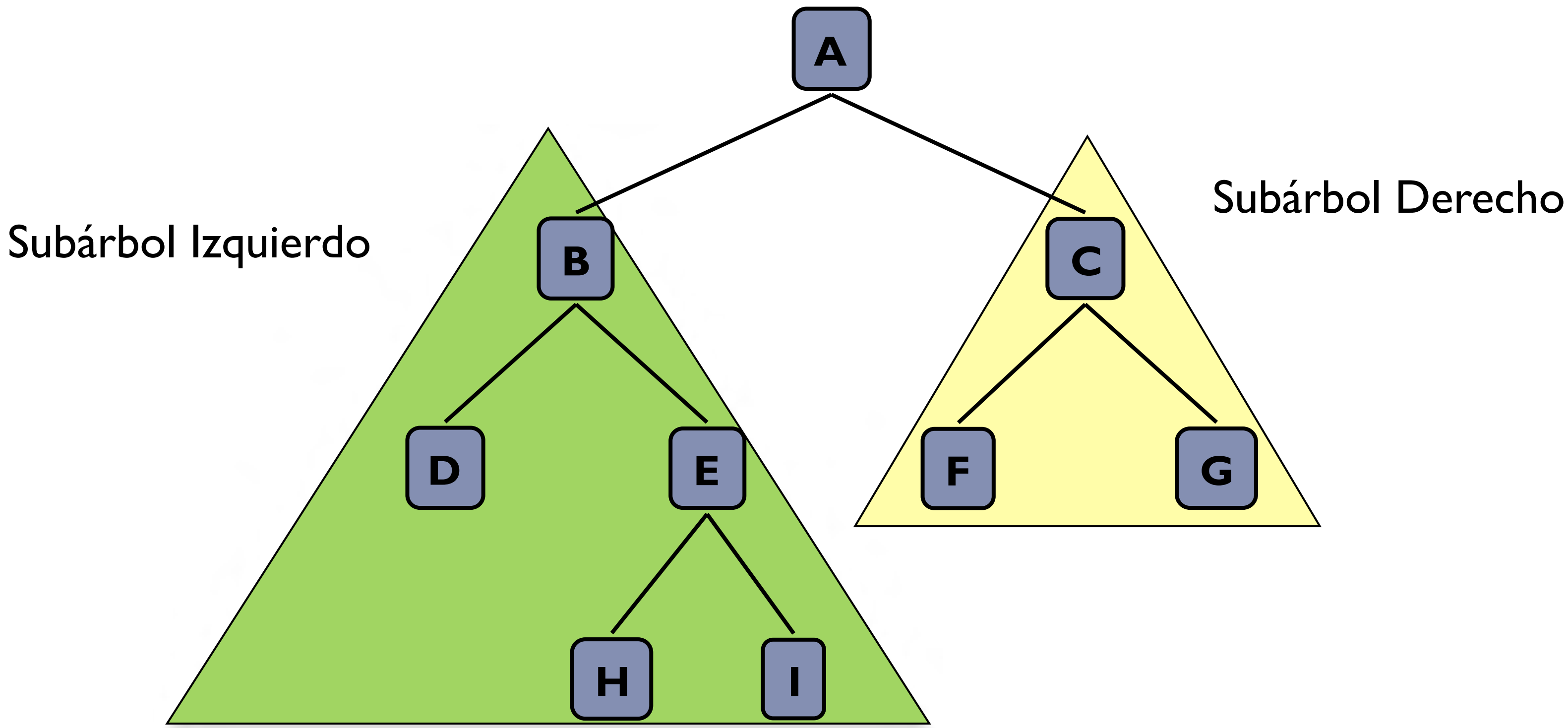
B

Índice

- Conceptos básicos
- TAD Árboles generales
- **TAD Árboles binarios**
- TAD Árboles binarios de búsqueda
- TAD Árboles B

Árboles Binarios

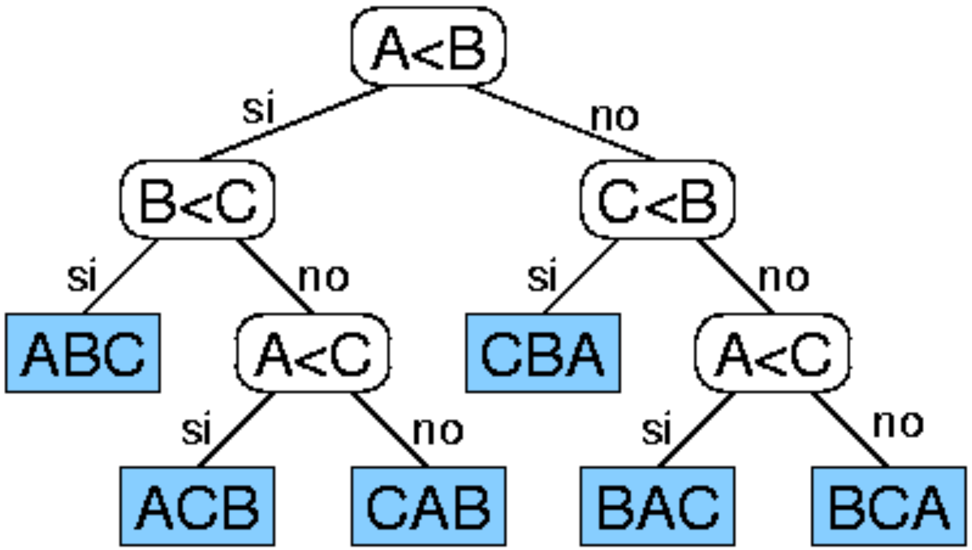
- Árbol de grado 2
  - Cada nodo tiene dos sub-árboles (pueden ser vacíos)



Árboles binarios: Aplicación

Ejemplo I: Árboles de decisión

- nodo interno: preguntas con respuesta si/no
- nodos hoja: decisiones

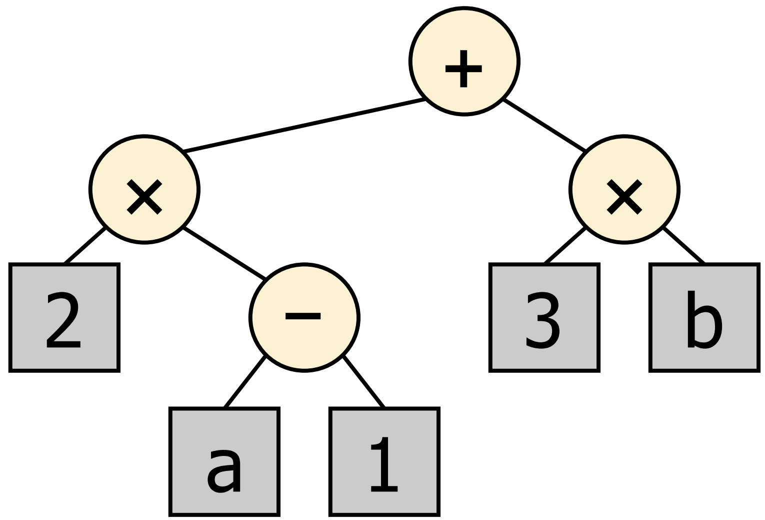


Ejemplo de un árbol de decisión para ordenar tres elementos A, B y C.

Árboles binarios: Aplicación

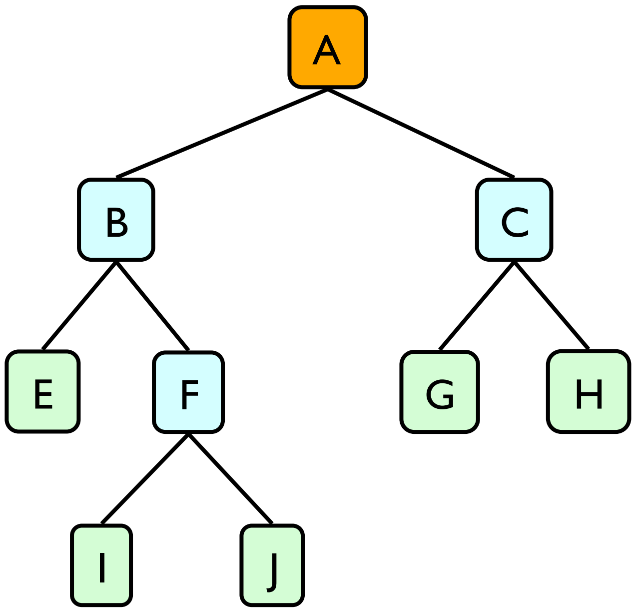
Ejemplo II: representar expresiones aritméticas

- nodo interno: operadores
- nodos hoja: operandos

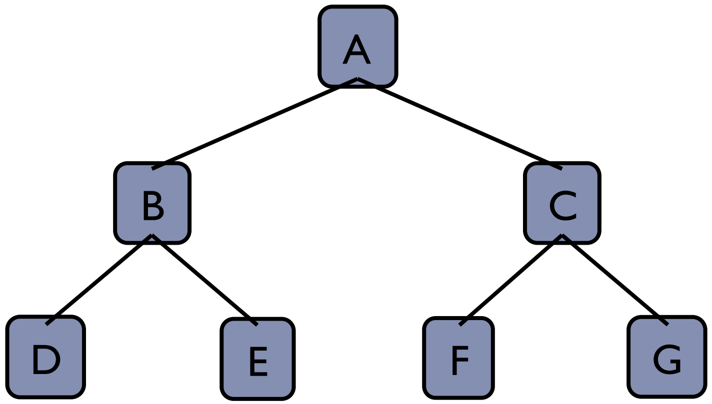


Árboles Binarios

- Es **completo** si todo nodo interno (no hoja) tiene dos descendientes

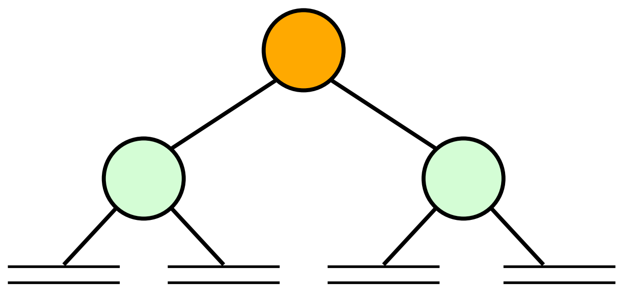


- Está **lleno** si es completo y además todas sus hojas están en el mismo nivel



Árboles binarios: propiedades

- **Notación**
  - n: número de nodos = 3
  - e: número de nodos hoja = 2
  - i: número de nodos internos =1
  - h: altura del árbol = 1



- **Propiedades**
  - $n = 2e-1 = 3$
  - $h \leq i \rightarrow 1 \leq 1$
  - $h \leq (n-1)/2 \rightarrow 1 \leq 1$
  - $e \leq 2^h \rightarrow 2 \leq 2$
  - $h \geq \log_2 e \rightarrow 1 \geq 1$
  - $h \geq \log_2(n+1)-1 \rightarrow 1 \geq 1$

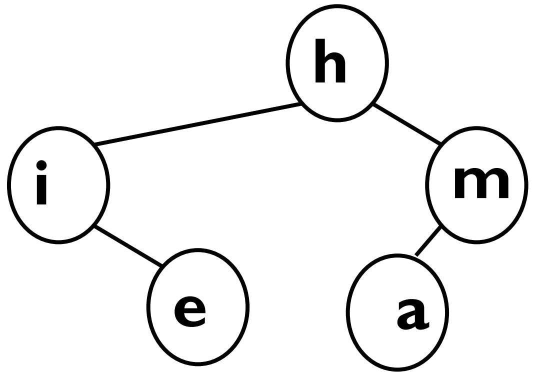
- **Si es completo:**
  - $e = i+1 \rightarrow 2 = 2$
  - $e \geq h+1 \rightarrow 2 \geq 2$

Árboles binarios: Recorridos

Recorrido PRE-ORDER

- primero se visita cada nodo, luego su subárbol izquierdo y finalmente el derecho (raiz, izq, der)
- Ejemplo:

pre-order: (h, i, e, m, a)

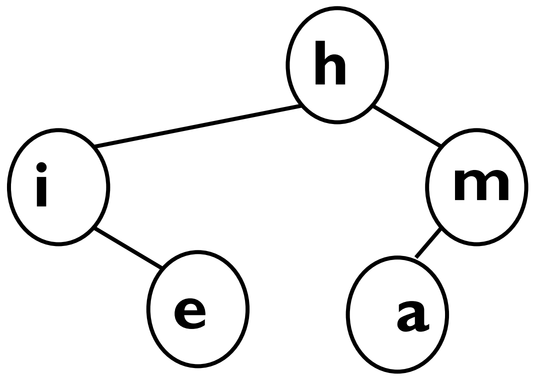


Árboles binarios: Recorridos

Recorrido POST-ORDEN

- cada nodo se visita después de visitar su subárbol izquierdo y después de visitar el derecho (izq, der, raiz)
- Ejemplo:

post-orden: (e, i, a, m, h)



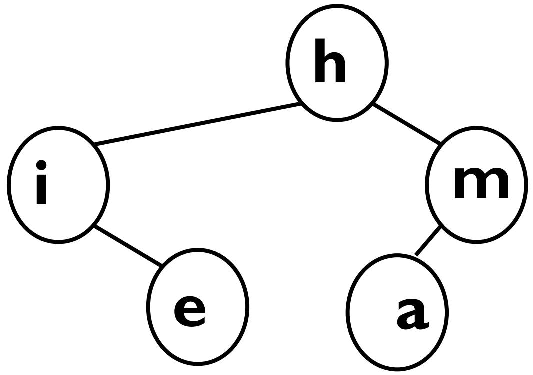
Árboles binarios: Recorridos

Recorrido POR NIVELES (LEVEL-ORDER)

- Se visitan los nodos en orden por nivel (en profundidad). Es decir, se visitan los nodos del mismo nivel de forma descendiente y de izquierda a derecha

- Ejemplo:

  Level-order: (h,i,m,e,a)

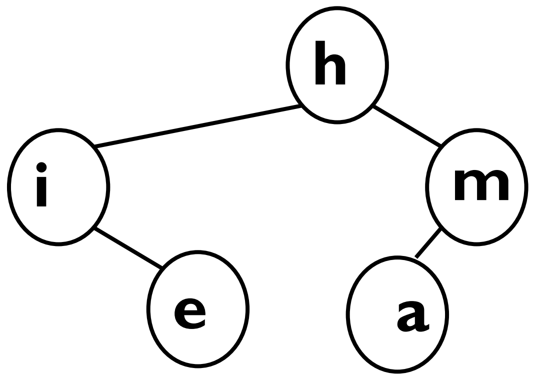


Árboles binarios: Recorridos

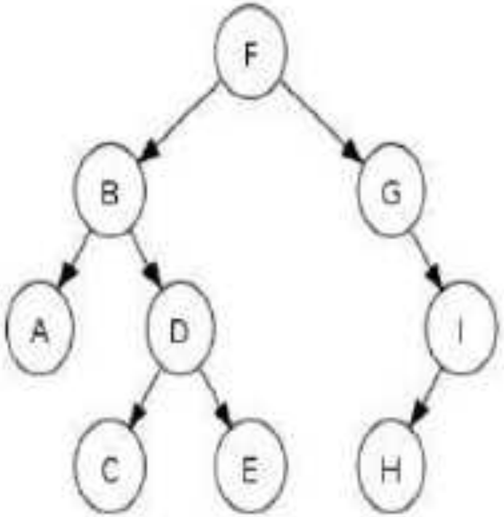
Recorrido IN-ORDER **(nuevo en árboles binarios)**

- cada nodo se visita tras visitar su subárbol izquierdo y antes de visitar el derecho (izq, raiz, der)
- Ejemplo:

in-order: (i, e, h, a, m)



Ejemplo Recorridos



In this binary search tree

- Preorder traversal sequence: F, B, A, D, C, E, G, I, H (root, left, right)
- Inorder traversal sequence: A, B, C, D, E, F, G, H, I (left, root, right); note how this produces a sorted sequence
- Postorder traversal sequence: A, C, E, D, B, H, I, G, F (left, right, root)
- Level-order traversal sequence: F, B, G, A, D, I, C, E, H

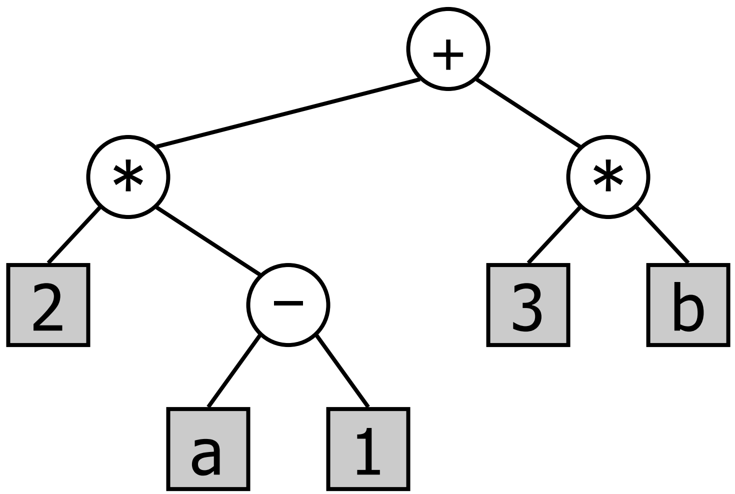
Árboles binarios: Ejercicio 3

El recorrido en "pre-orden" de un árbol binario es: EXAMFUN y en "in-orden" MAFXUEN, donde cada carácter es un nodo.

- Dibujar el árbol binario.
- Dar el recorrido en post-orden.
- Dar el recorrido por niveles del árbol.

Árboles binarios: Ejercicios de recorridos

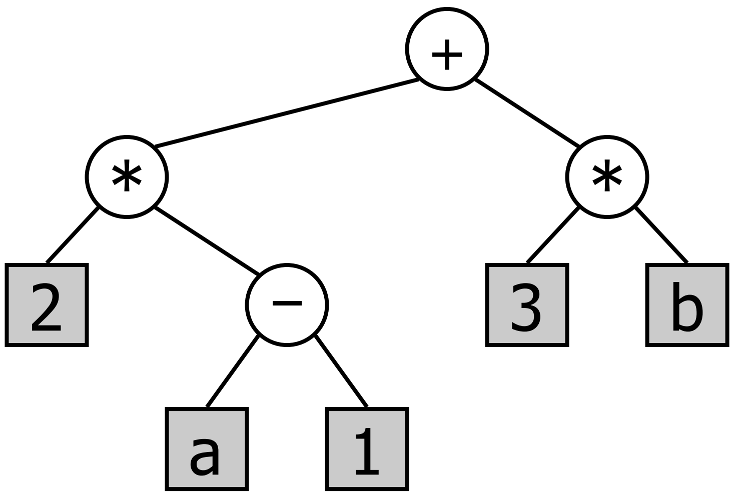
- **Ejemplo: expresiones aritméticas**
  - pre-order: notación prefija (polaca)
  - in-order: notación normal (sin paréntesis)
  - post-order: notación polaca inversa



pre-order: + * 2 - a 1 * 3 b (raiz, izq, der)
in-order: 2 * a - 1 + 3 * b (izq, raiz, der)
post-order: 2 a 1 - * 3 b * + (izq, der, raiz)

Ejercicio. Paréntesis en expresión matemática

- Dada una expresión matemática en un TAD árbol binario
  - escribir (de forma teórica) el algoritmo para que el recorrido en in-order incluya los paréntesis

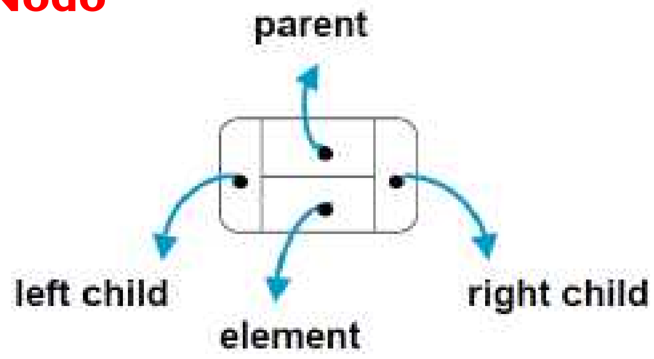


in-order: 2 * (a - 1)+ 3 * b

Nota: es interesante poner todos los paréntesis, pero si sólo se añaden los necesarios para su correcto cálculo, mejor.

Árboles binarios: Implementación

Nodo



Árbol