Algoritmos y Estructuras de Datos I

Universidad Católica del Uruguay

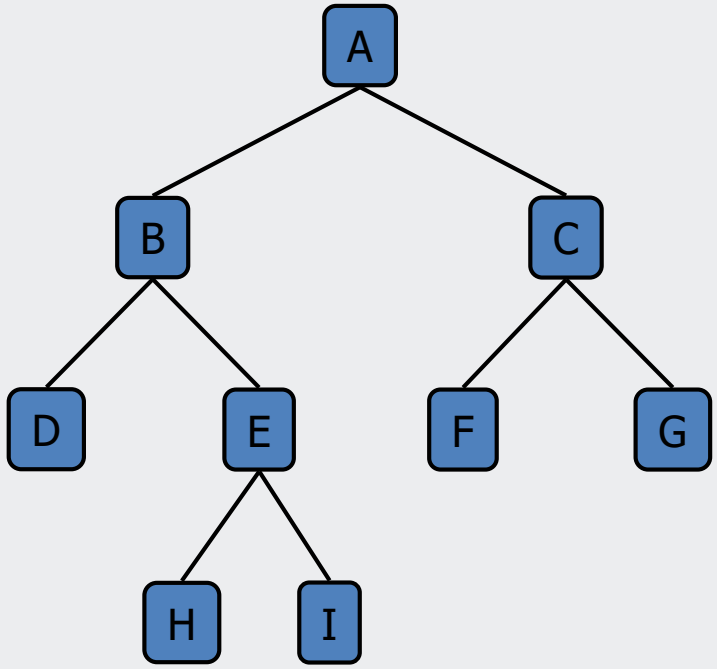
Arboles Binarios – I

Resumen de clase 3 de mayo de 2021

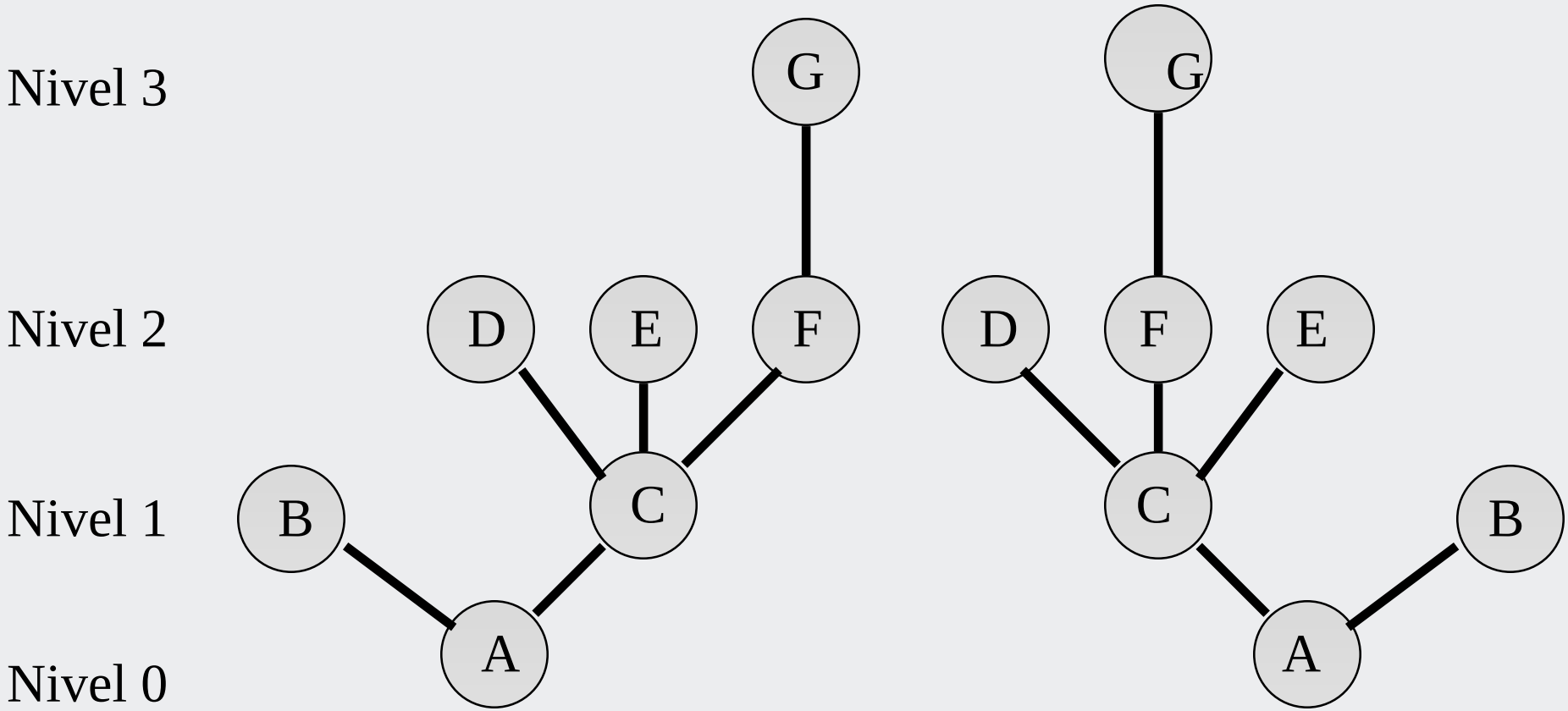
Arboles Binarios

- Un Arbol Binario es un Arbol con las siguientes propiedades:
  - Cada nodo interno tiene como máximo dos hijos (exactamente **dos** para árboles binarios completos)
  - Los hijos de un nodo son un par ordenado
- Los hijos de un nodo interno se denominan Hijo Izquierdo e Hijo Derecho
- Definiciones alternativas:
  - Un árbol con un sólo nodo, o un árbol cuya raíz tiene un par ordenado de hijos, cada uno de los cuales es a su vez un árbol binario
  - Conjunto finito de nodos, que puede estar vacío o consistir de una raíz y dos árboles binarios disjuntos, llamados subárbol izquierdo y derecho de la raíz.

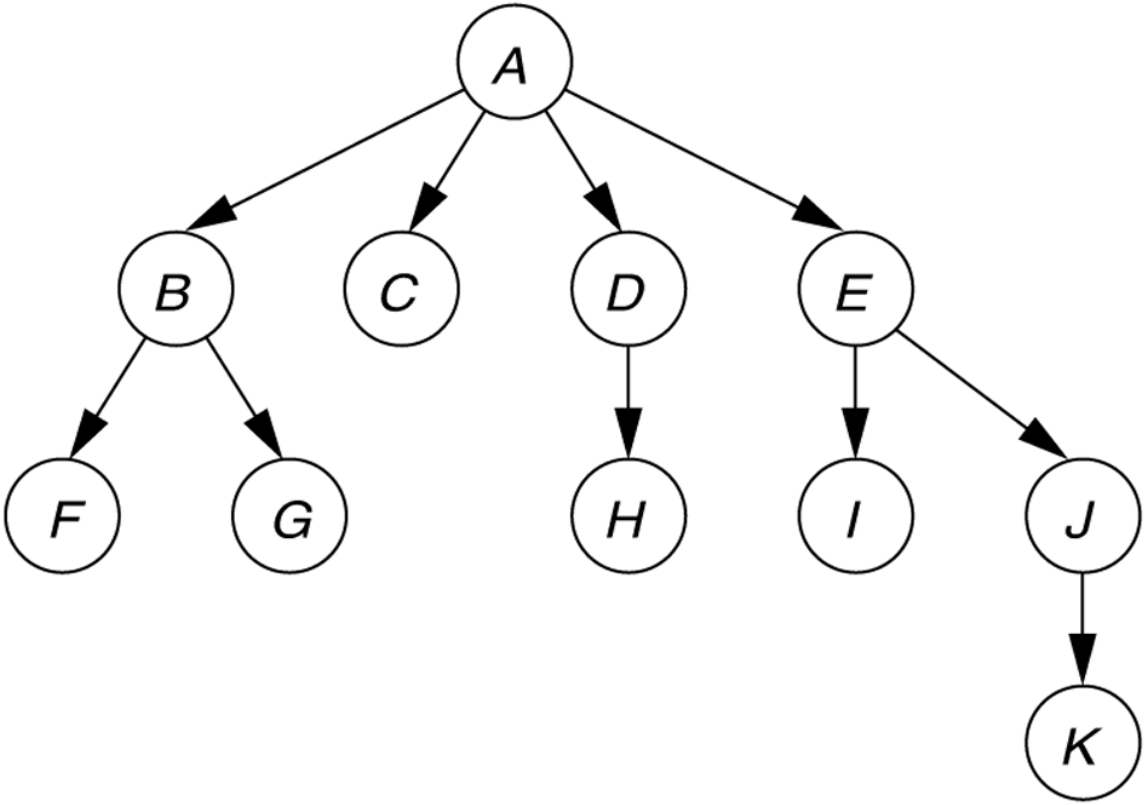
- **Applicaciones:**
  - Expresiones aritméticas
  - Procesos de decisión
  - Búsqueda



ARBOLES



Altura y profundidad o nivel



Nodo	Altura	Nivel
A	3	0
B	1	1
C	0	1
D	1	1
E	2	1
F	0	2
G	0	2
H	0	2
I	0	2
J	1	2
K	0	3

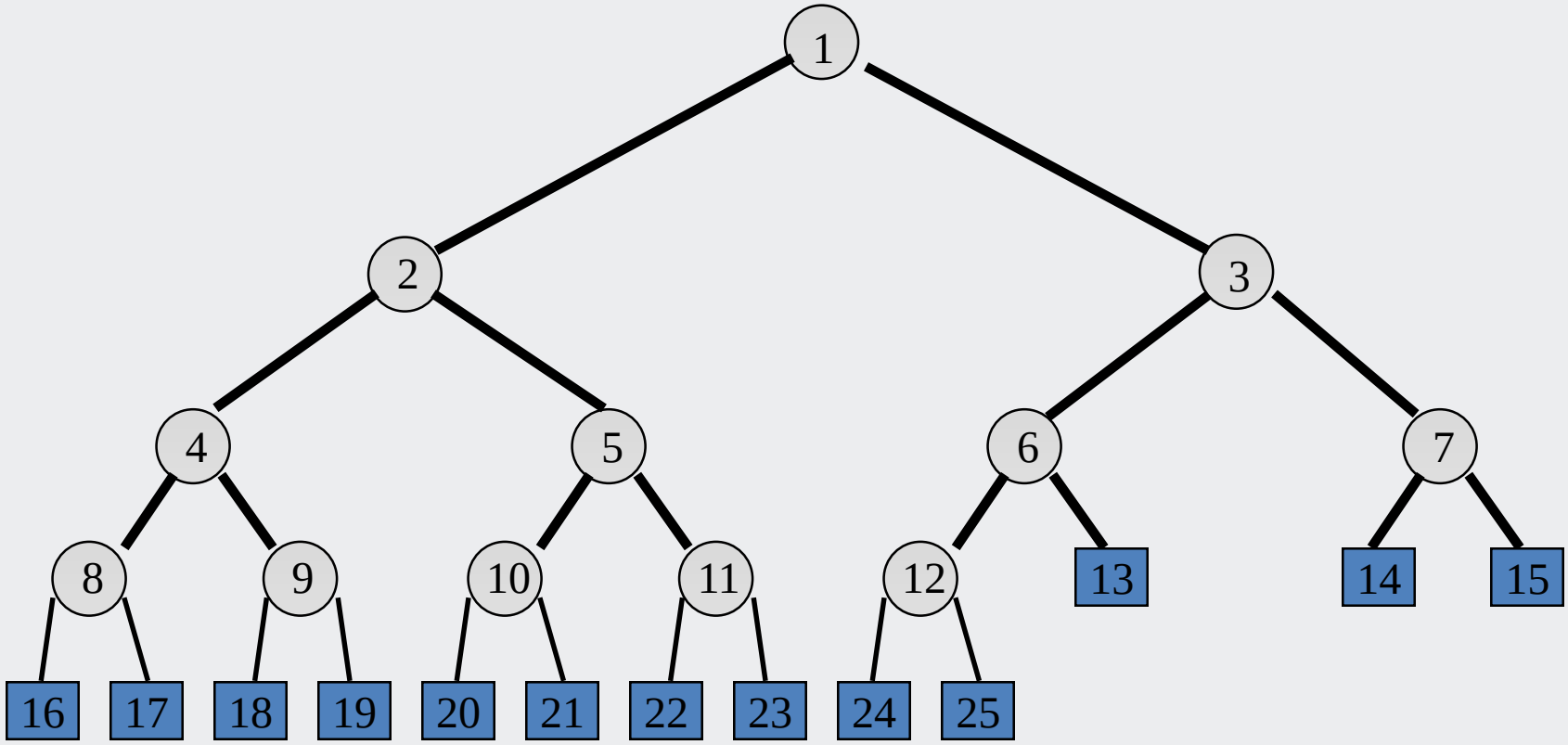
ARBOLES: Representación

- La raíz arriba, las hojas abajo.
- Se dice que cada raíz es el “padre” de las raíces de sus subárboles, los cuales son llamados “hermanos”.
- Los subárboles son llamados “hijos” de su “padre”.
- La raíz del árbol total no tiene padre.
- Ancestros y descendientes.

Lleno y completo

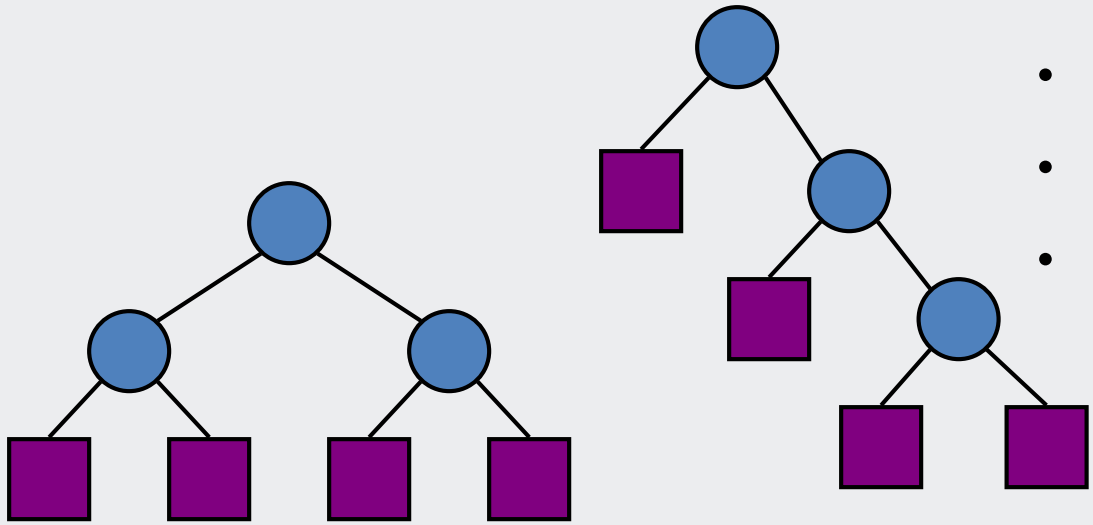
- Arbol binario Lleno
- Arbol binario completo

ARBOL BINARIO COMPLETO



Propiedades de los Arboles Binarios llenos

- **Notación**
  - n número de nodos
  - e número de nodos externos
  - i número de nodos internos
  - h altura



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

Arboles y recursividad

Operaciones en arboles binarios

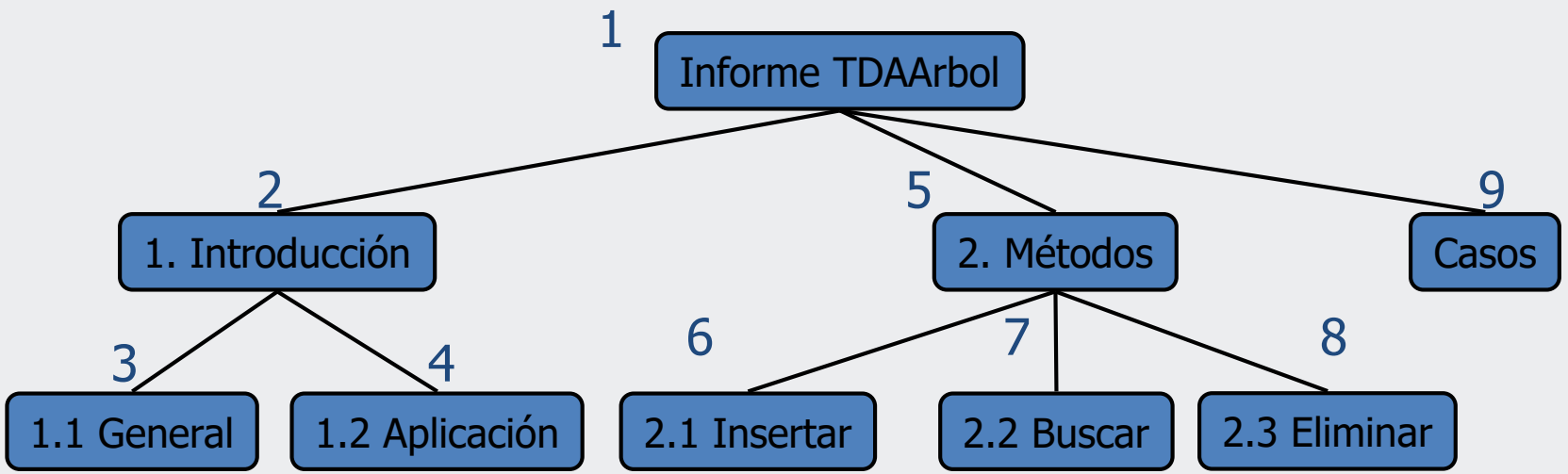
Universidad Católica del Uruguay

- Operaciones básicas.
  - Inserción, búsqueda, eliminación
- Recorridos
  - Preorden
  - Postorden
  - inorden

Recorrida en Preorden

- En un recorrido en preorden, un nodo es visitado antes que sus descendientes
- Aplicación: imprimir un documento estructurado

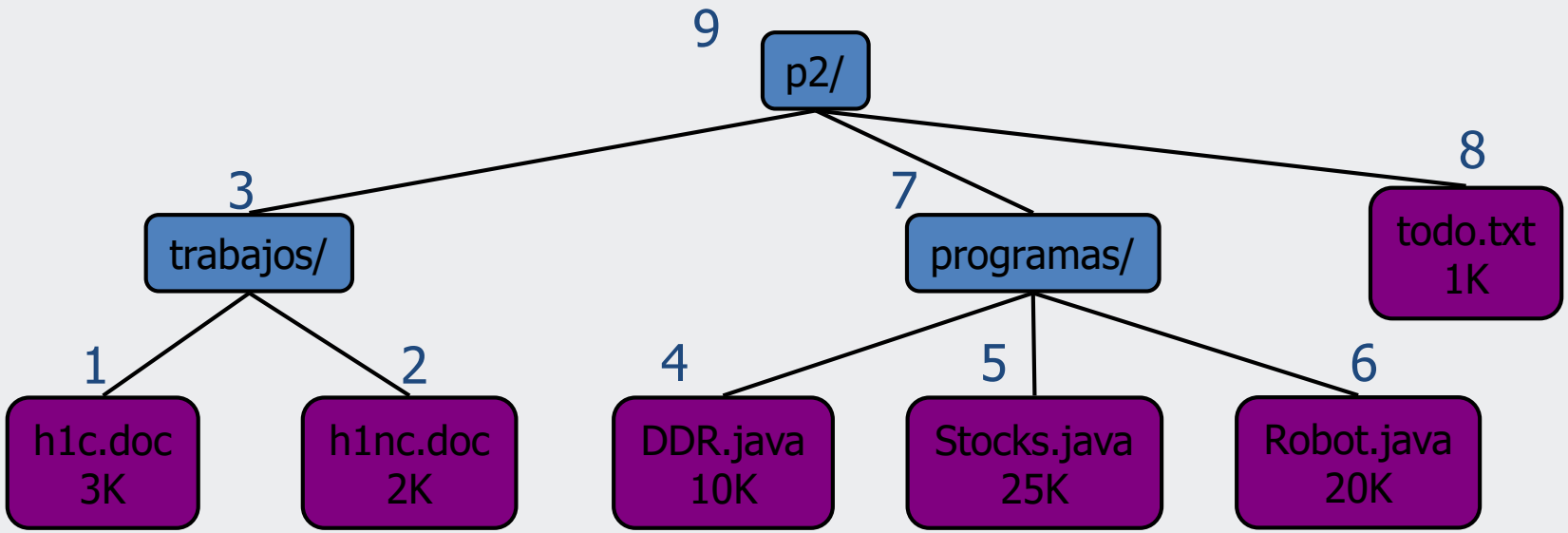
Algoritmo ***preOrden*****(*****v*****)**
  visitar**(*****v*****)**
  Para cada hijo ***w*** de ***v***
    preorden **(*****w*****)**



Recorrida en Postorden

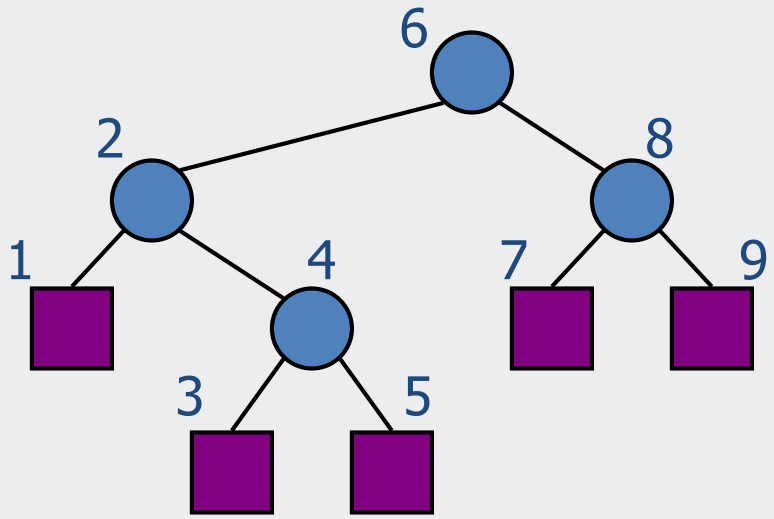
- En un recorrido en postorden, un nodo es visitado después de sus descendientes
- Aplicación: calcular el espacio usado por archivos en un directorio y sus subdirectorios

Algoritmo *postOrden*(*v*)
  Para cada hijo *w* de *v*
    postOrden (*w*)
  visitar(*v*)



Recorrido en Inorden de un árbol binario

- En un recorrido en inorden el nodo es visitado después que su subárbol izquierdo y antes que su subárbol derecho



Algoritmo ***TAB.InOrden***

Si *raiz no es nula*

raiz.InOrden

Algoritmo ***TNodoAB.InOrden***

Si *tiene HijoIzquierdo*

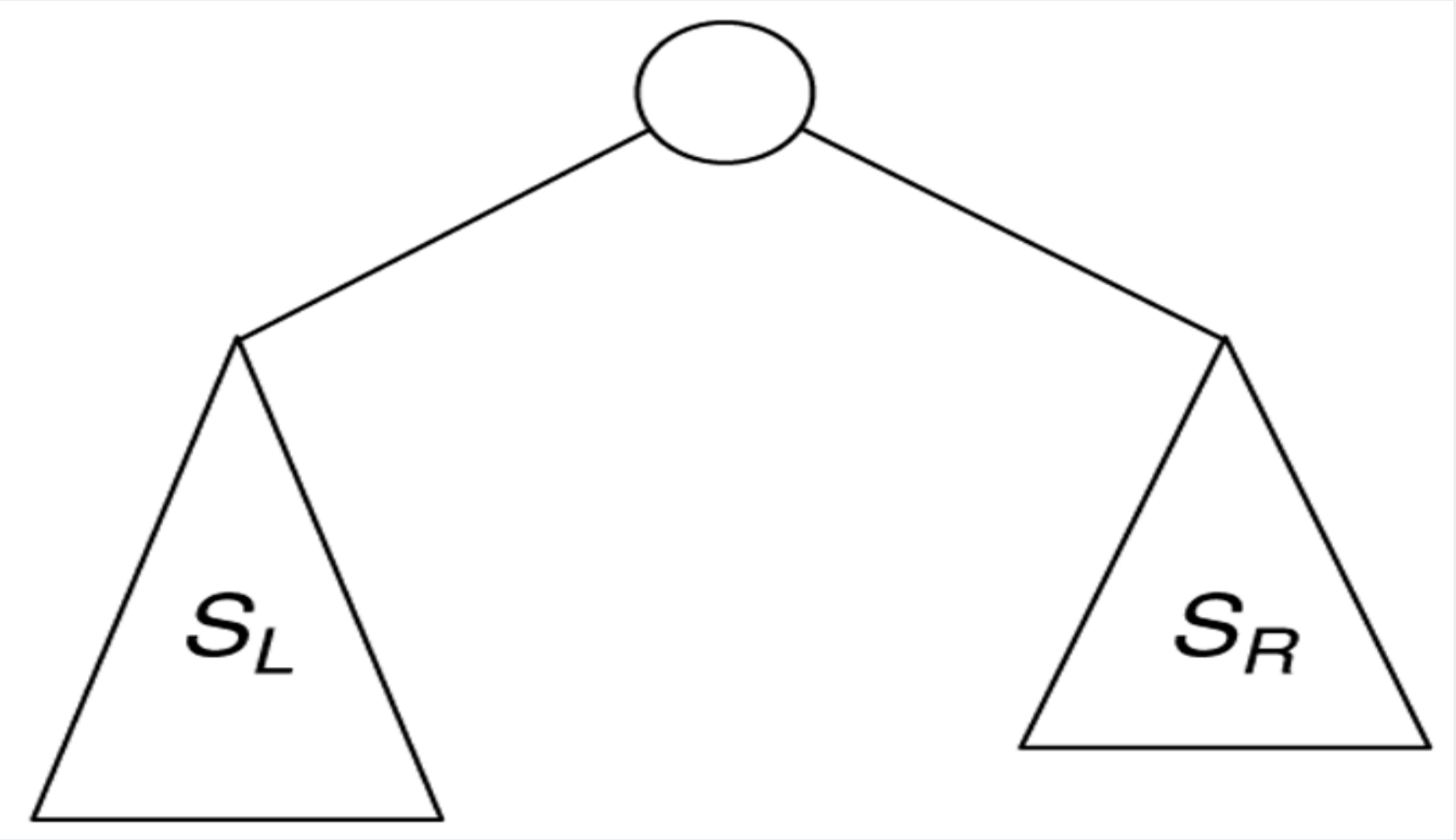
HijoIzquierdo.inOrden

visitar(v)

Si *tiene HijoDerecho*

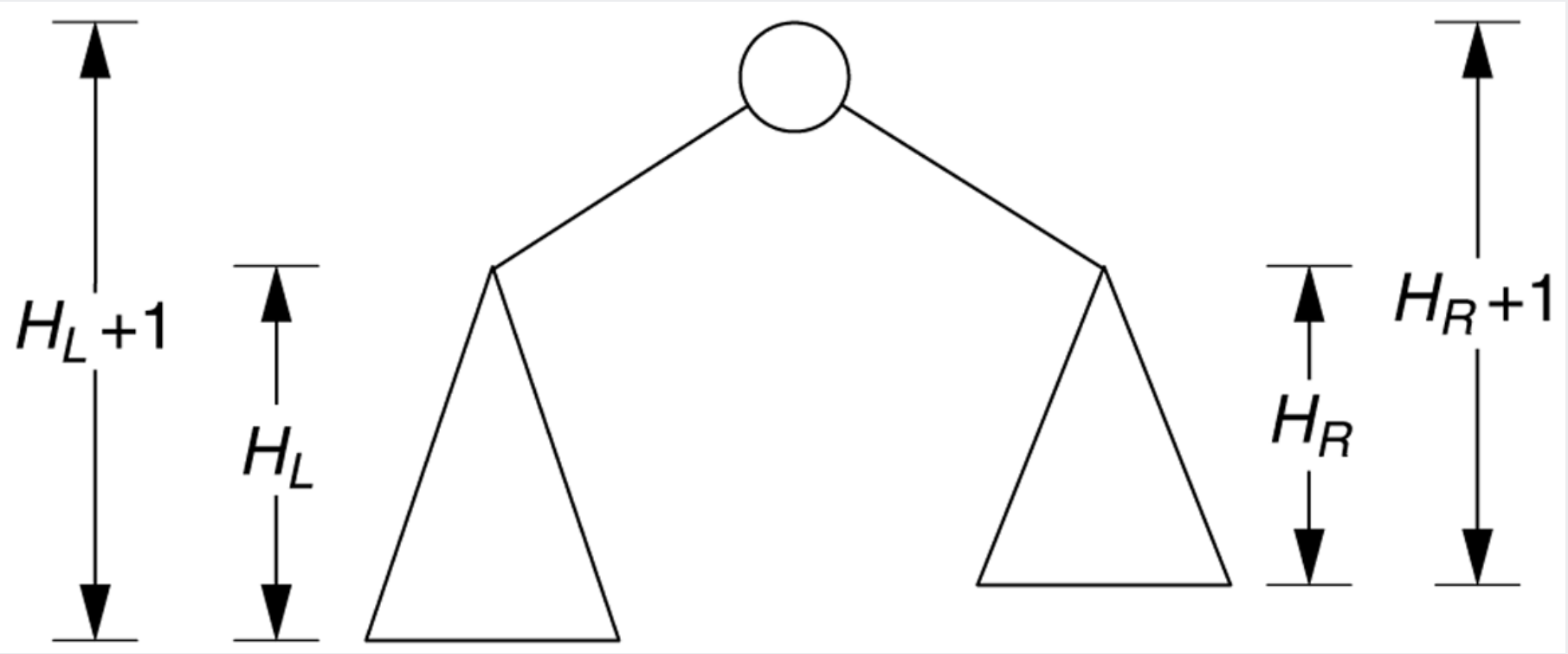
HijoDerecho. inOrden

Vista recursiva usada para calcular el tamaño de un árbol $ST = SL + SR + 1$



Vista recursiva usada para calcular la altura de un nodo:

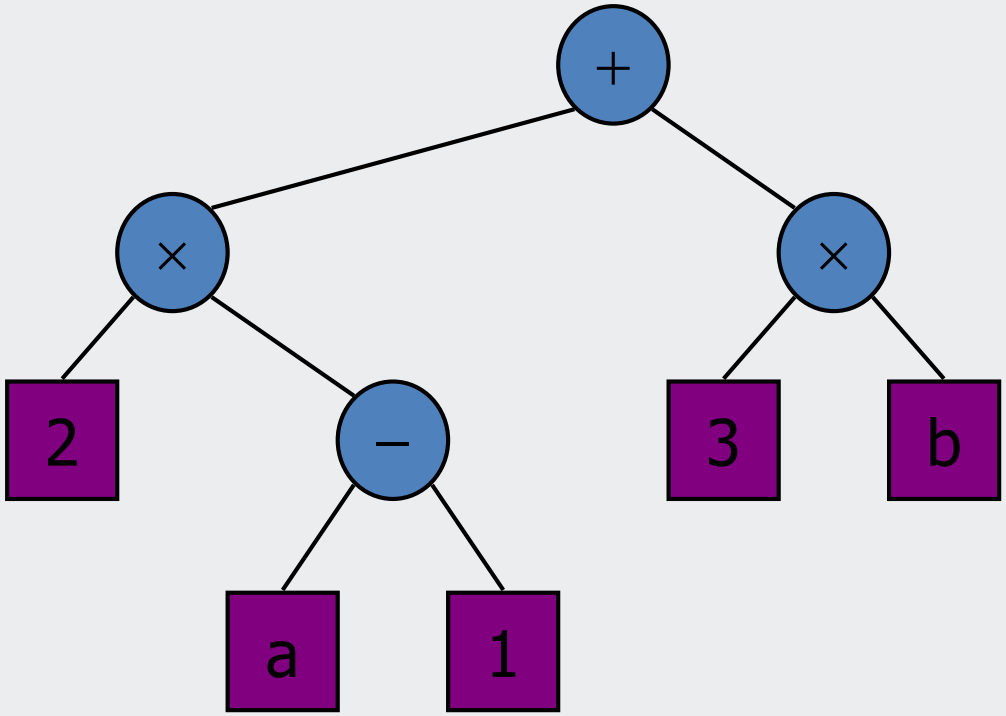
HT = Max (HL + 1, HR + 1)



Arbol de Expresión Aritmética

- Arbol binario asociado con una expresión aritmética:
  - Nodos internos: operadores
  - Nodos externos: operandos

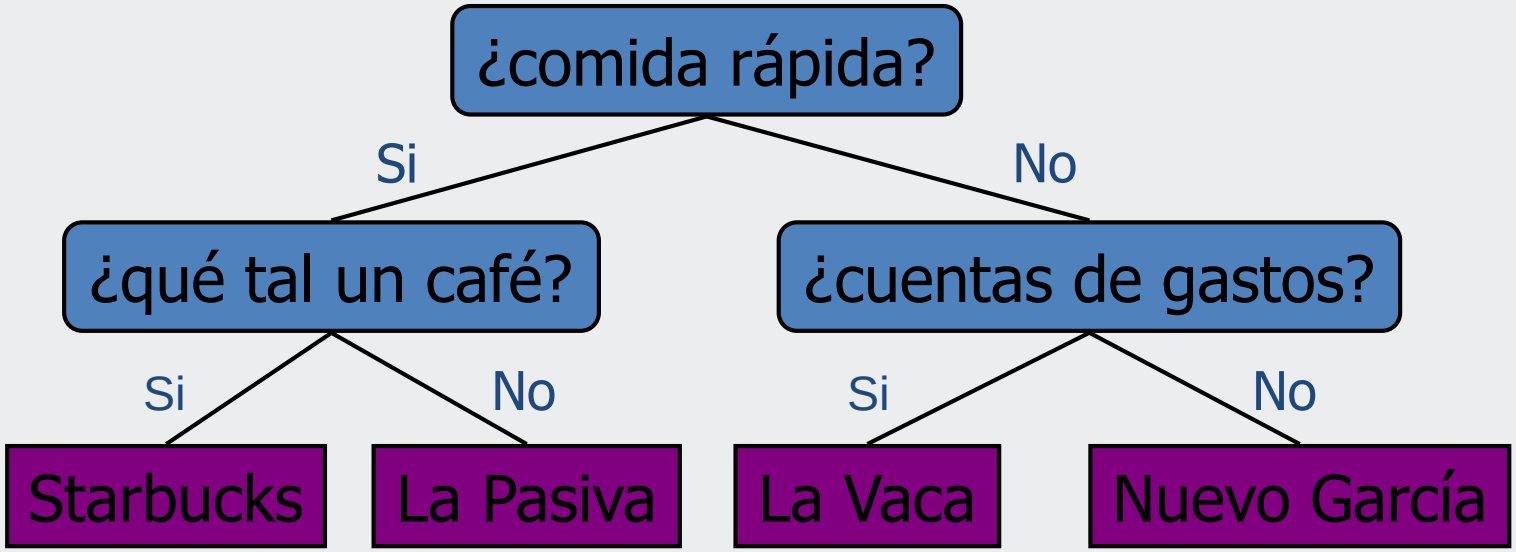
- Ejemplo: árbol de expresión aritmética para la expresión $(2 \times (a - 1) + (3 \times b))$
- Notación polaca: **+x2-a1x3b**



Aplicación de Arbol Binario: Arbol de Decisión

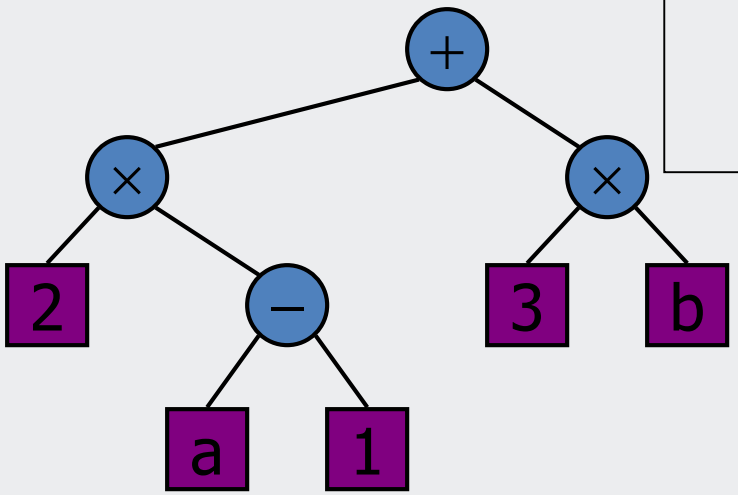
- Arbol binario asociado con un proceso de decisión
  - Nodos internos : preguntas con respuestas si / no
  - Nodos externos : decisiones

- Ejemplo: decisión de cena



Impresión de expresiones aritméticas

- Especialización del recorrido en inorden
  - Imprimir el operando u operador al visitar el nodo
  - Imprimir "(" antes de recorrer el subárbol izquierdo
  - Imprimir ")" después de recorrer el subárbol derecho



Algoritmo *TNodoAB.printExpression*

Si *tieneHijoIzquierdo*

imprimir("(")

HijoIzquierdo.printExpression

imprimir

Si *tieneHijoDerecho*

HijoDerecho .printExpression

imprimir(")")

$$((2 \times (a - 1)) + (3 \times b))$$

Evaluar expresiones aritméticas

- Especialización del recorrido en postorden
  - Método recursivo que retorna el valor de un subárbol
  - Al visitar un nodo interno, combina los valores de los subarboles

Algoritmo *TNodoAB.evalExpr*

Si *esHoja*

Devolver *elemento*

sino

x ← *HijoIzquierdo .evalExpr*

y ← *HijoDerecho.evalExpr*

◊ ← operador contenido

devolver *x* ◊ *y*