Algoritmos
Estructuras de Datos I

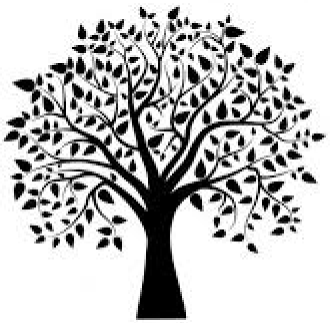
Facultad de Ciencias Exactas y Tecnología

Universidad Nacional de Tucumán

2024

Universidad Nacional de Tucumán
facet
Facultad de Ciencias Exactas y Tecnología

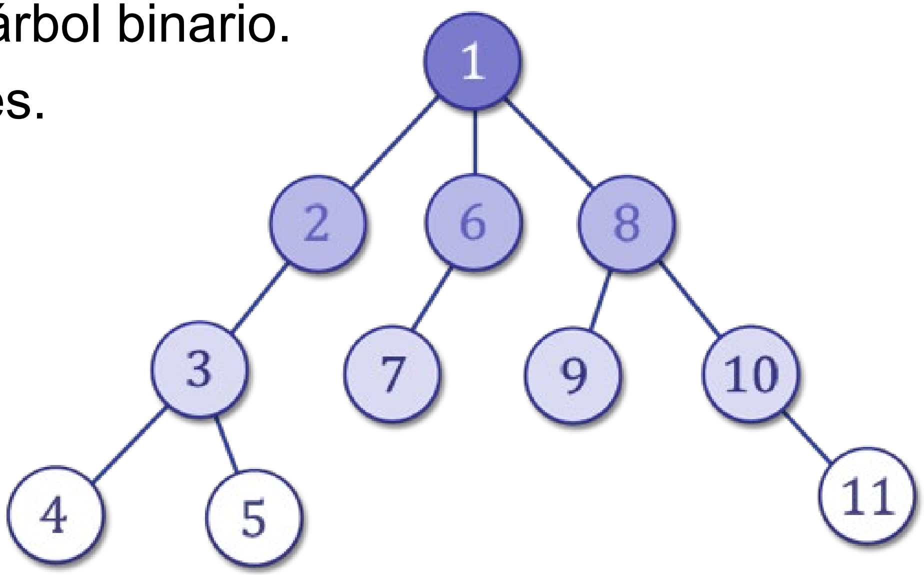
Árbol



Unidad III

Tipos de datos no lineales. ARBOL

- Definición de árbol. Terminología.
- El tipo abstracto de datos árbol.
- El tipo abstracto de datos árbol binario.
- Distintas implementaciones.
- Tipos de arboles binarios
- Aplicaciones.



ARBOL

Un árbol, a diferencia de las estructuras de datos vistas hasta ahora, es una estructura ***no lineal*** que aparece frecuentemente en algoritmos en diferentes áreas en Ciencias de la Computación.

El árbol es una estructura de datos muy importante en informática, con muchas aplicaciones en modelos de la vida real.

Un árbol impone una ***estructura jerárquica*** en una colección de objetos.

ARBOL

Se pueden usar para:

- Representar: arboles genealógicos, expresiones aritméticas, estructuras sintácticas de programas fuente y de fórmulas matemáticas, caminos en un juego, la organización de competencias deportivas, la organización de archivos en los sistemas operativos, etc
- Organizar: índices, las carpetas y archivos en los dispositivos de almacenamiento, la información para realizar búsquedas eficientes en sistemas de bases de datos.
- Y mucho más.

ARBOL

DEFINICION:

Un árbol es una colección de objetos denominados ***nodos*,** uno de los cuales se distingue como ***raíz***, junto con una relación (de paternidad) que impone una estructura jerárquica sobre los nodos.

Un nodo, así como en el caso de un elemento de una estructura lineal, puede ser de cualquier tipo (item).

ARBOL

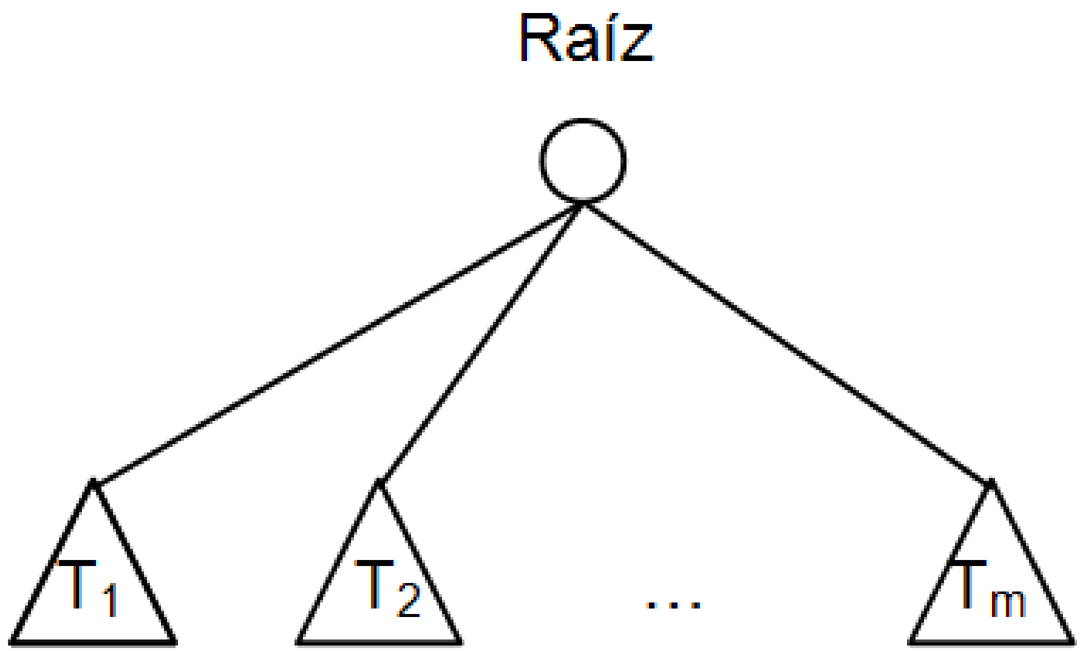
DEFINICION FORMAL:

Un árbol puede definirse de manera recursiva como un conjunto finito T de nodos tales que:

- Hay un nodo especial llamado la ***raíz*** del árbol, un nodo solo es, por si mismo, un árbol.
- Los nodos restantes están agrupados en m≥0 conjuntos disjuntos: T1,T2,…,Tm y cada uno de esos conjuntos es a su vez un árbol. Los arboles T1,T2,…,Tm se llaman ***subárboles*** de la raíz.
- Es conveniente incluir en la definición el ***árbol nulo*** o ***vacío***, un árbol tal que no tiene nodos (se representa por Λ).

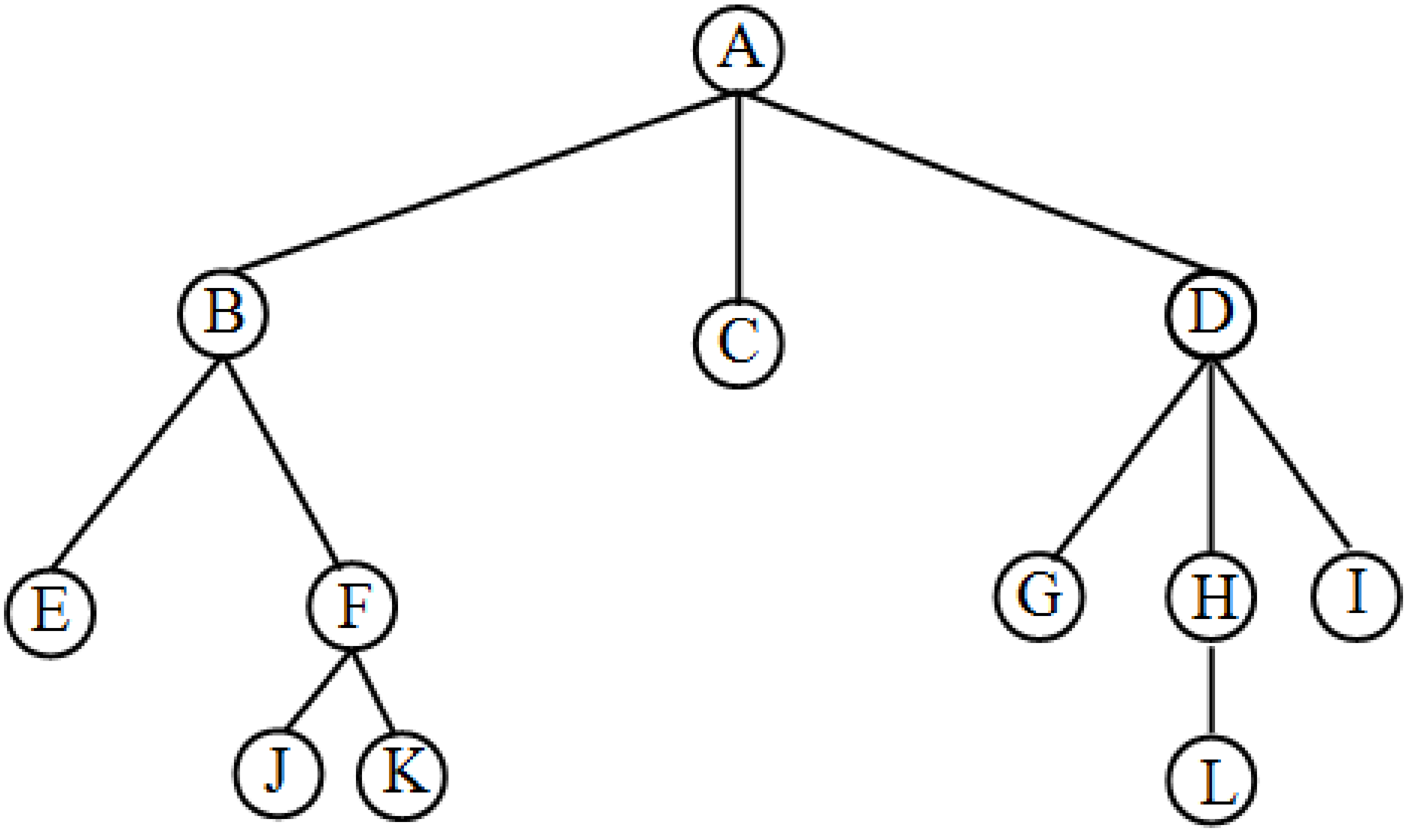
ARBOL

Arbol T:



Representación de Árbol

Grafo



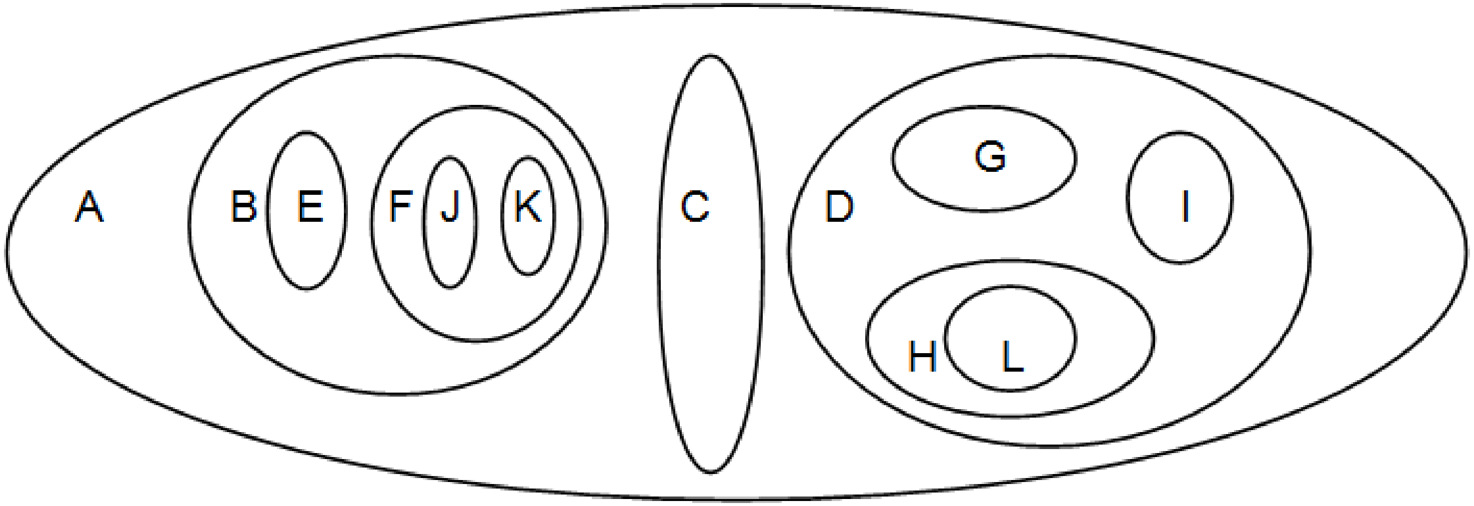
Representación de Árbol

Indentado

```
A
.........B
....................E
....................F
............................J
............................K
.........C
.........D
....................G
....................H
............................L
....................I
```

Representación de Árbol

Conjunto



Paréntesis

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

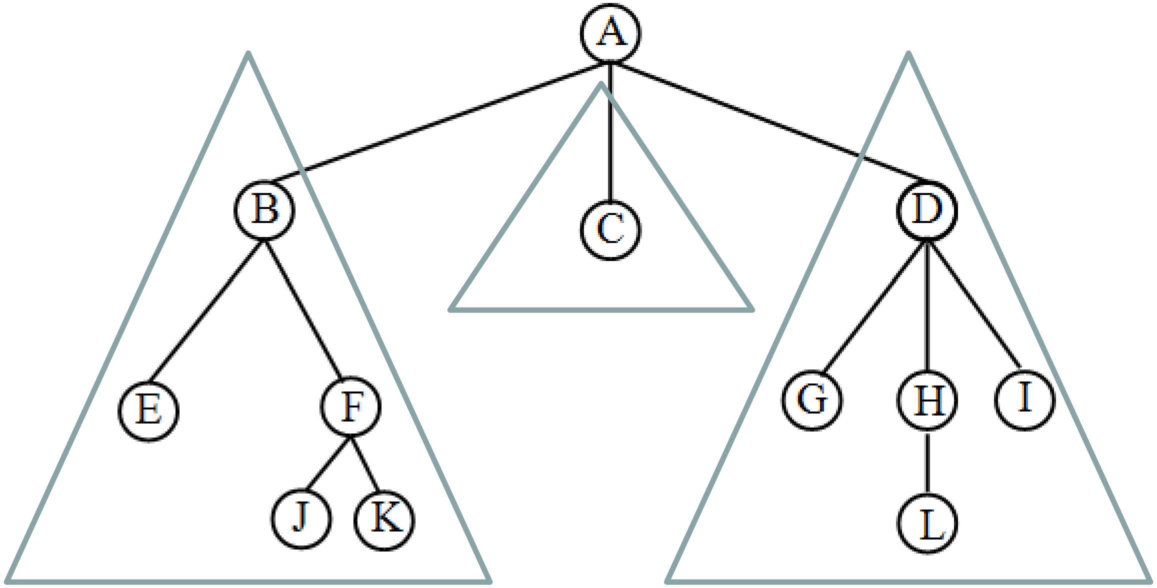
TERMINOLOGIA

- Padre, Hermanos, Hijos.
- Camino, Longitud de camino.
- Nivel
- Antecesor, Descendiente.
- Antecesor y Descendiente propio o directo.
- Hoja o nodo terminal, Nodo interior.
- Altura de un árbol.
- Profundidad de un nodo, Profundidad de un árbol.
- Grado de un nodo, Grado de un árbol.
- Bosque.

TERMINOLOGIA

- **Padre**: cada nodo es padre de sus subárboles.
- **Hermanos**: los subárboles que dependen de un mismo nodo son hermanos entre si.
- **Hijos**: cada subárbol es hijo de su raíz

Ejemplo:



TERMINOLOGIA

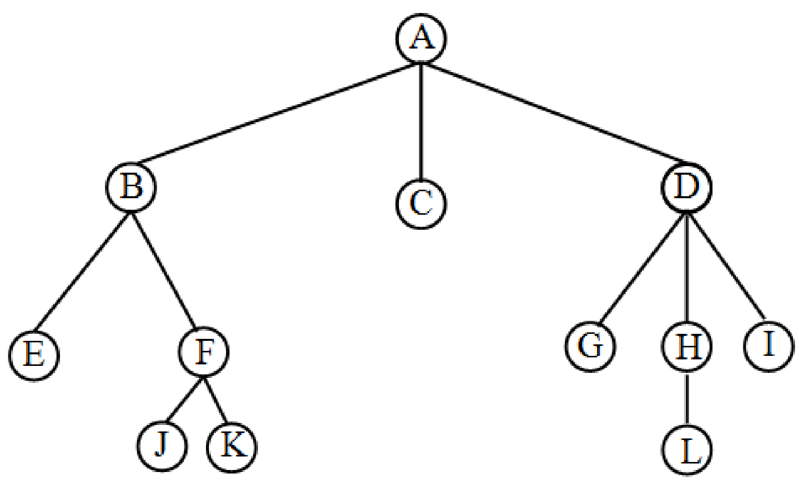
- **Camino**: si $n_1,n_2,\ldots,n_k$ es una secuencia de nodos de un árbol tal que n_i es padre de n_{i+1} para $1 \leq i < k$, entonces la secuencia se denomina *camino* del nodo n_1 al nodo n_k.
- Hay un solo camino desde la raíz del árbol a cada nodo.
- **Longitud de camino**: es el numero de nodos del camino menos uno.
- Hay un camino de longitud cero de cualquier nodo a si mismo.

Ejemplos:

<A,B,E> <D,G>

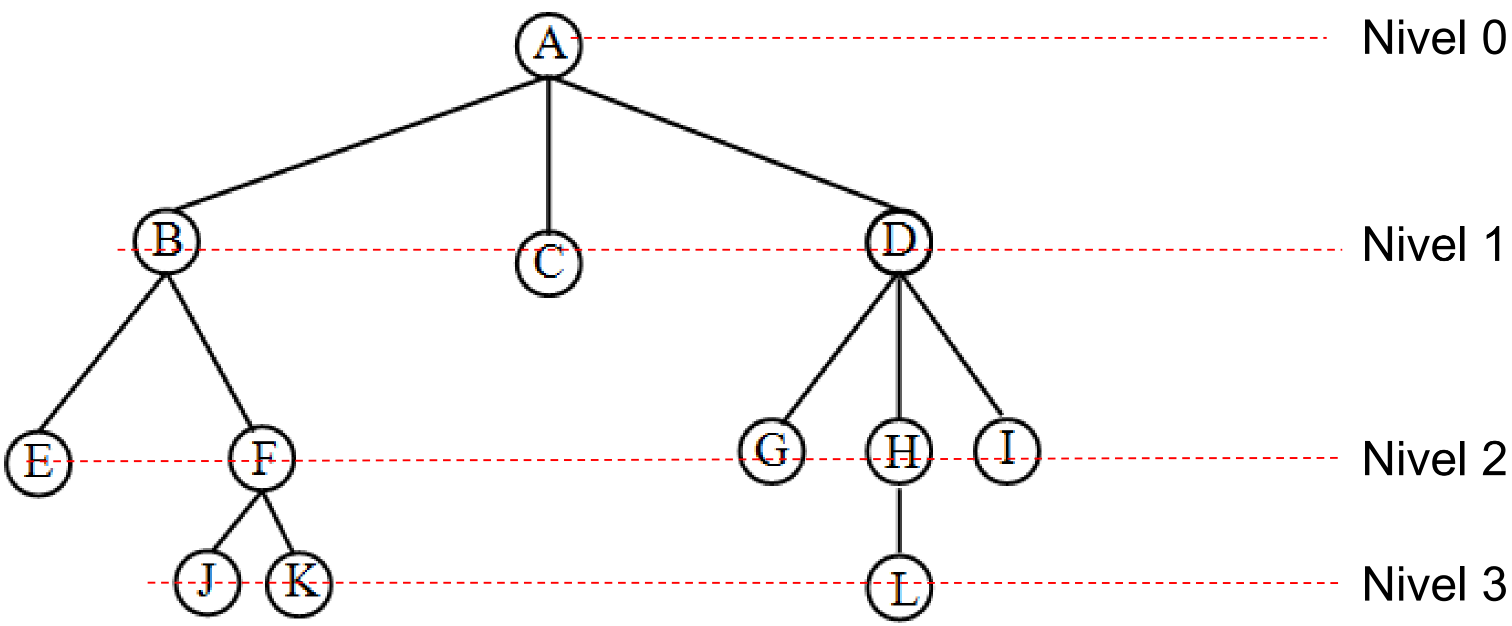
<A,B,F,J> <F,K>

<A,D,H,L> <D,H,L>



TERMINOLOGIA

- **Nivel**: el nivel de un nodo es el numero de nodos del camino que va desde la raíz a dicho nodo menos uno. La raíz principal esta en el nivel 0, sus hijos en nivel 1 y así.

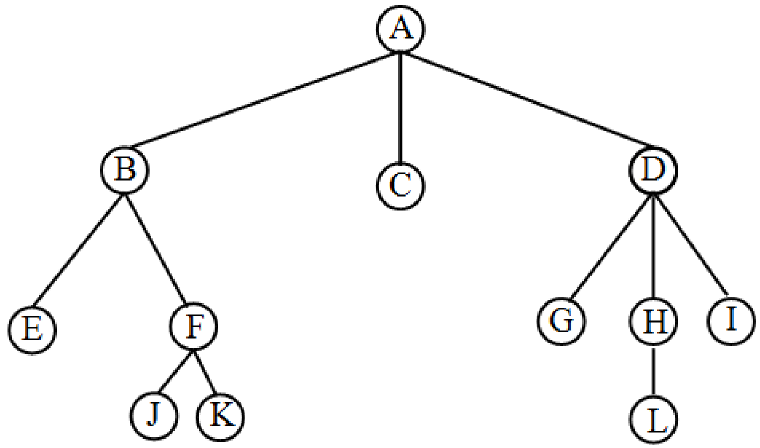


TERMINOLOGIA

- ***Antecesor/Descendiente***:

Si existe un camino de un nodo *A* a otro nodo *B*, entonces *A* es un ***antecesor*** de *B* y *B* es un ***descendiente*** de *A*. Los antecesores de un nodo son todos los nodos a lo largo del camino desde la raíz a ese nodo.

- ***Antecesor propio o directo***: un antecesor propio de un nodo en nivel (**i**) esta en el nivel (**i-1**) y sus ***descendientes propios o directos*** están en nivel (**i+1**).



TERMINOLOGIA

- **Hoja o nodo terminal**: es un nodo sin descendientes propios.
- **Nodo interior**: es un nodo que no sea terminal.

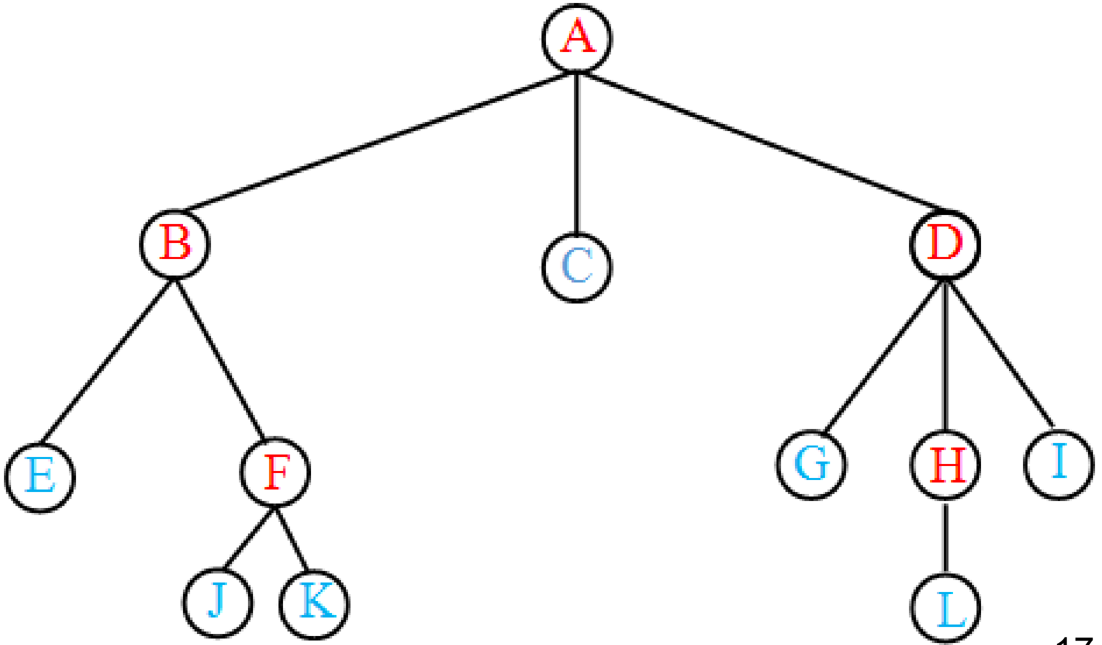
Ejemplo:

Nodos interiores:

A,B,D,F,H

Nodos hojas:

C,E,G,I,J,K,L



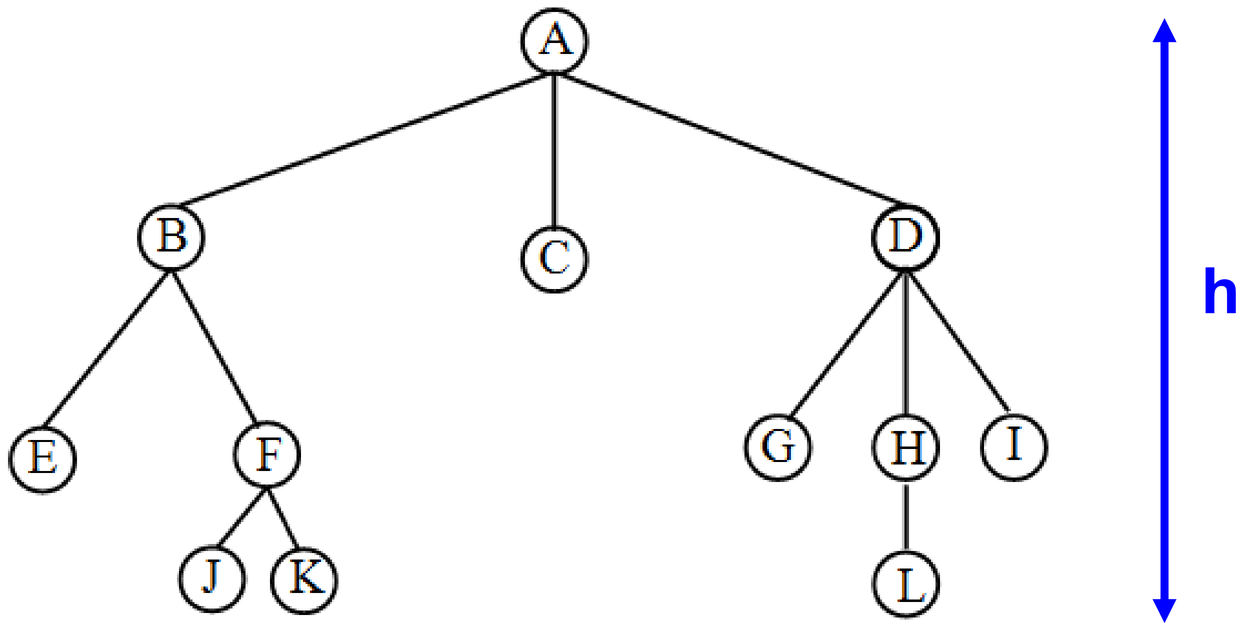
TERMINOLOGIA

- **Altura de un árbol**: es la longitud del camino mas largo de la raíz a una hoja.

Ejemplo:

Altura(T)= **3**

h(T)= **3**



TERMINOLOGIA

- **Profundidad de un nodo**: es la longitud del camino único desde la raíz a ese nodo.
- **Profundidad de un árbol:** es el máximo nivel de cualquier hoja del árbol, esto es el camino mas largo desde la raíz a una hoja. Coincide con altura.

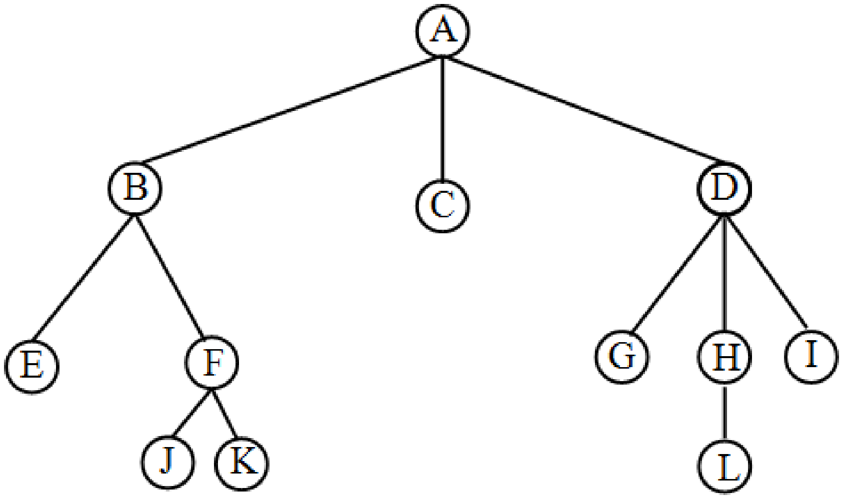
Ejemplos:

Altura(T)=Profundidad(T)=**3**

Profundidad(A)=**0**

Profundidad(B)=**1**

Profundidad(E)=**2**



TERMINOLOGIA

- **Grado de un nodo**: es el numero de sus descendientes propios. Las hojas tienen grado cero.
- **Grado de un árbol**: es el máximo de los grados de todos los nodos del árbol.

Ejemplos:

Grado(A)=**3**

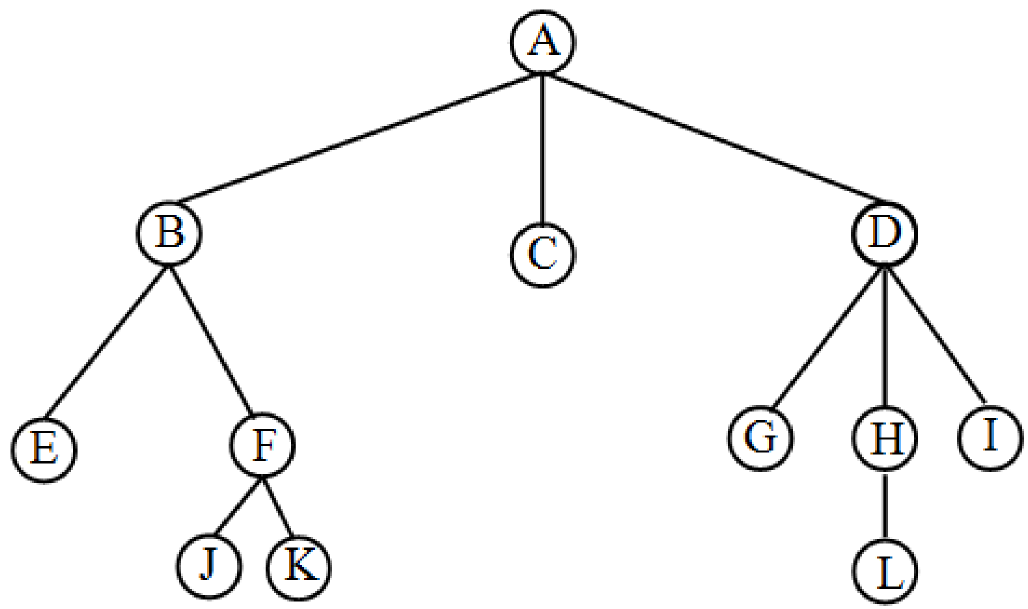
Grado(B)=**2**

Grado(H)=**1**

Grado(C)=**0**

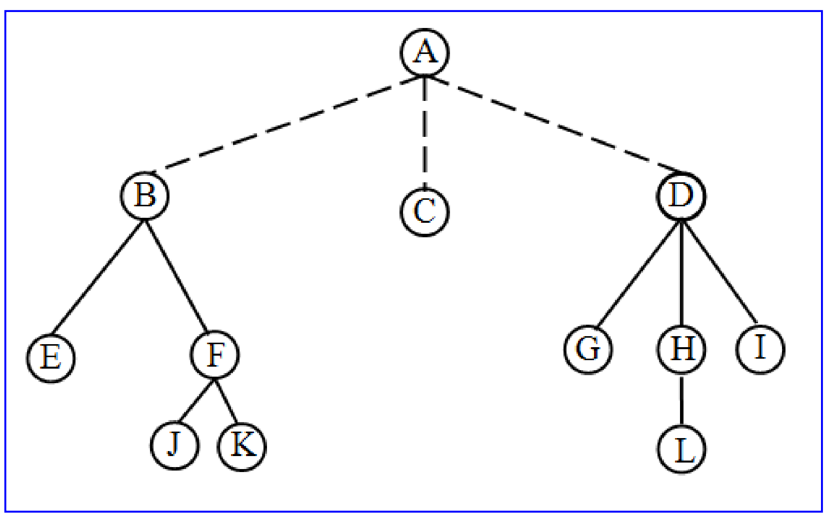
…

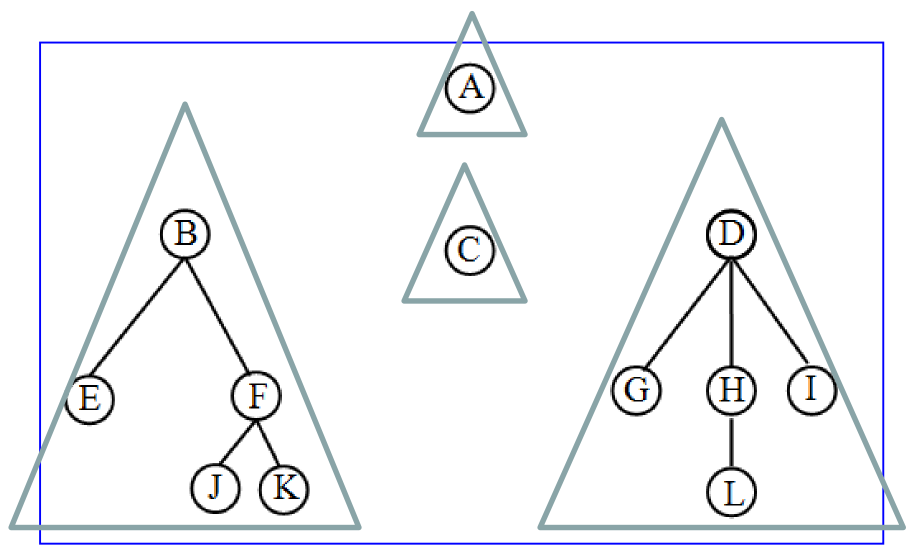
Grado (T) = **3**



TERMINOLOGIA

- **Bosque**: un conjunto de n≥0 arboles disjuntos es un bosque.
- Si en un árbol se suprime la raíz y las aristas que las unen con los subárboles se obtiene un bosque.

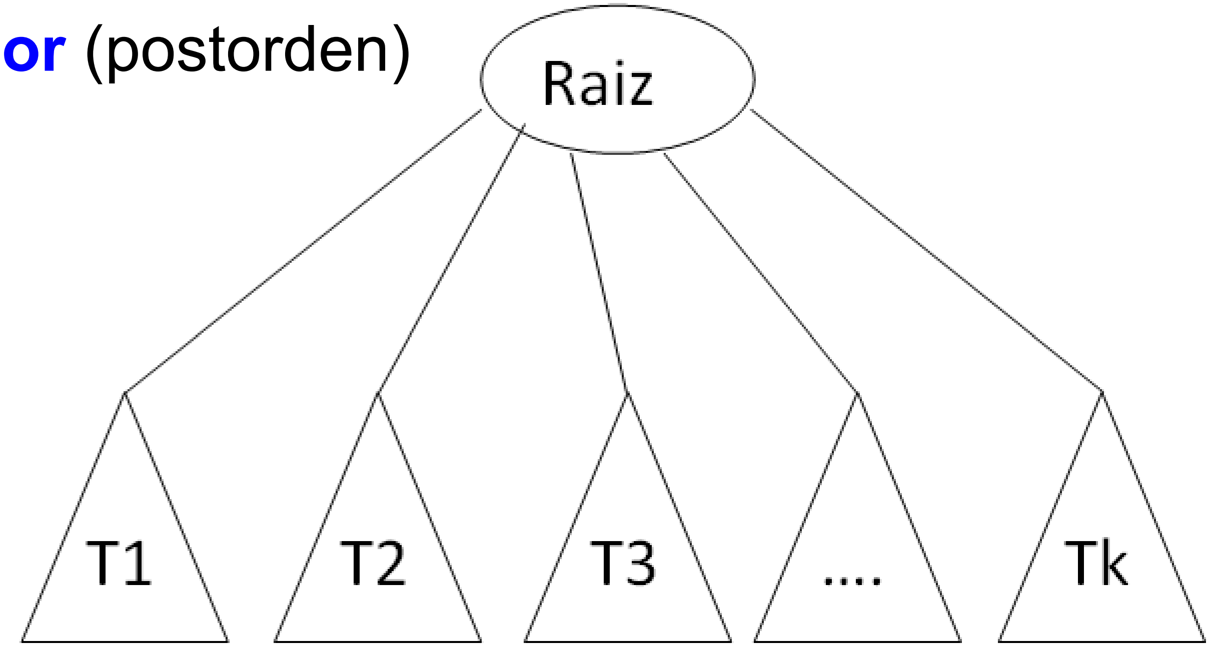




Recorrido de Arbol

Los recorridos sistemáticos son:

- Orden **previo** (preorden),
- Orden **simétrico** (enorden)
- Orden **Posterior** (postorden)
- Por **nivel**



Recorrido de Arbol

- Si T es un ***árbol vacío***, entonces la listavacia es el listado en ordenes previo, simétrico y posterior.

Opre ≡ < >

Osim ≡ < >

Opost ≡ < >

Recorrido de Arbol

- Si T contiene ***un solo nodo***, entonces ese nodo constituye el listado de los nodos de T en los ordenes previo, simétrico y posterior.

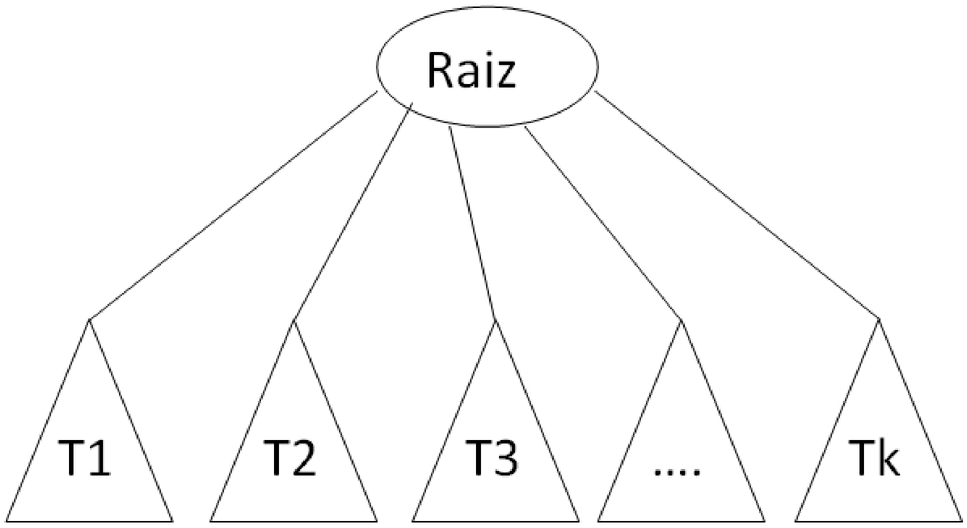
Opre ≡ <Raiz>

Osim ≡ <Raiz>

Opost ≡ <Raiz>

Recorrido de Arbol

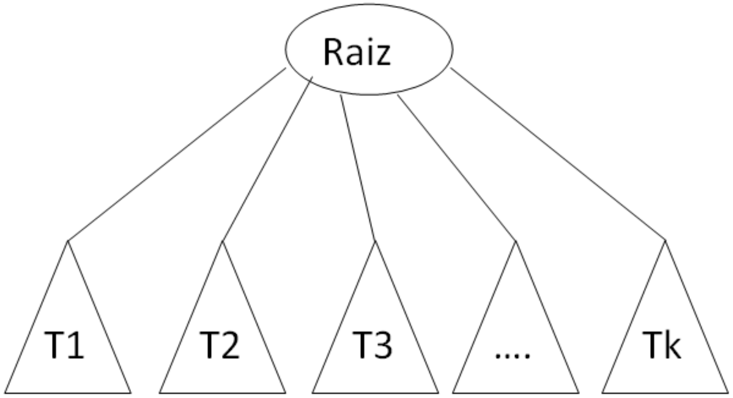
- Si no es ninguno de los casos anteriores entonces sea T un árbol con Raíz y subárboles: $T_1, T_2, \ldots, T_k$



Recorrido de Arbol

- El listado en **orden previo** de los nodos de T esta formado por la Raíz de T, seguida de los nodos de T_1 en orden previo, luego los nodos de T_2 en orden previo y así sucesivamente hasta los nodos de T_k en orden previo.

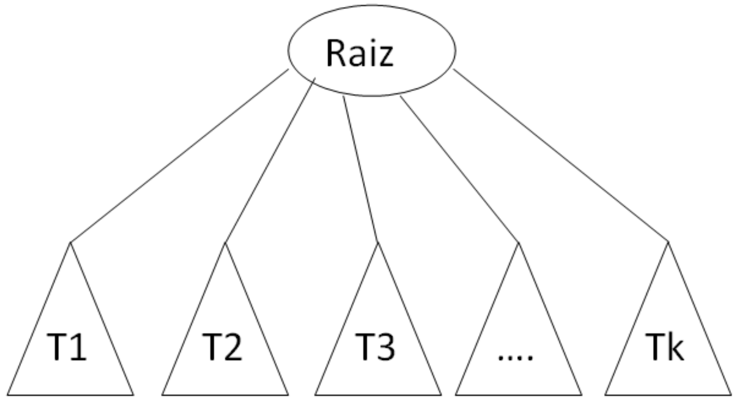
Opre $\equiv$ < Raiz, Opre(T_1), Opre(T_2),…,Opre(T_k) >



Recorrido de Arbol

- El listado en **orden simétrico** de los nodos de T esta formado por los nodos de T_1 en orden simétrico, seguidos de la Raiz y luego los nodos de $T_2,\ldots,T_k$ con cada grupo de nodos en orden simétrico.

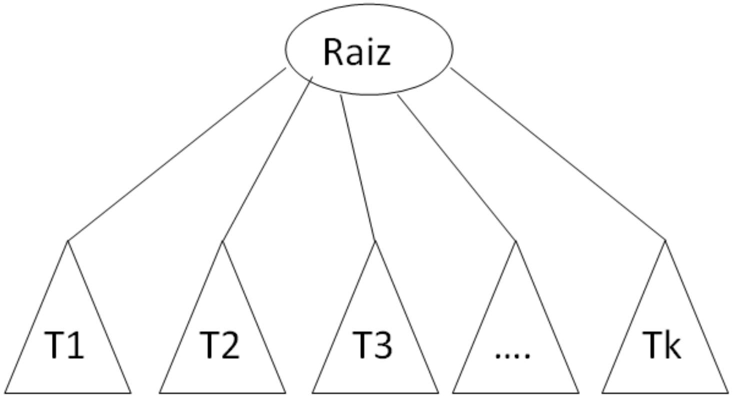
Osim $\equiv$ < Osim(T_1), Raiz, Osim(T_2),…,Osim(T_k) >



Recorrido de Arbol

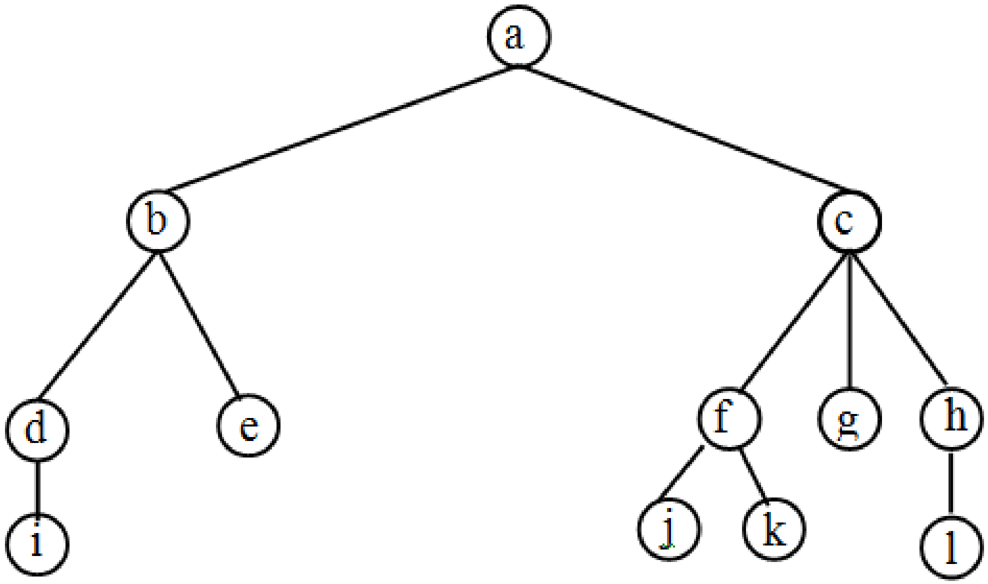
- El listado en **orden posterior** de los nodos de T tiene los nodos de T_1 en orden posterior, luego los nodos de T_2 en orden posterior y así sucesivamente hasta los nodos de T_k en orden posterior y por ultimo la Raíz.

Opost $\equiv$ < Opost(T_1), Opost(T_2),…,Opost(T_k) , Raiz >



Recorrido de Arbol

Ejemplo:



- Listado preorden ≡ **< a, b, d, i, e, c, f, j, k, g, h, l >**
- Listado enorden ≡ **< i, d, b, e, a, j, f, k, c, g, l, h >**
- Listado postorden ≡ **< i, d, e, b, j, k, f, g, l, h, c, a >**
- Listado por nivel ≡ **< a, b, c, d, e, f, g, h, i, j, k, l >**

ARBOL (ITEM)

Especificación algebraica - OPERACIONES:

A) Sintaxis:

ARBOLVACIO	:	→ ARBOL
ESARBOLVACIO	:	ARBOL → BOOLEAN
CONSTRUIR	:	ITEM X ARBOL X ARBOL → ARBOL
PRIMERHIJO	:	ARBOL → ARBOL
PROXHERMANO	:	ARBOL → ARBOL
RAIZ	:	ARBOL → ITEM U {indefinido}

ARBOL (ITEM)

B) Semántica: $\forall$ a,b $\in$ ARBOL , $\forall$ i $\in$ ITEM

ESARBOLVACIO(ARBOLVACIO) $\equiv$ TRUE

ESARBOLVACIO(CONSTRUIR(i,a,b)) $\equiv$ FALSE

PRIMERHIJO(ARBOLVACIO) $\equiv$ ARBOLVACIO

PRIMERHIJO(CONSTRUIR(i,a,b)) $\equiv$ a

PROXHERMANO(ARBOLVACIO) $\equiv$ ARBOLVACIO

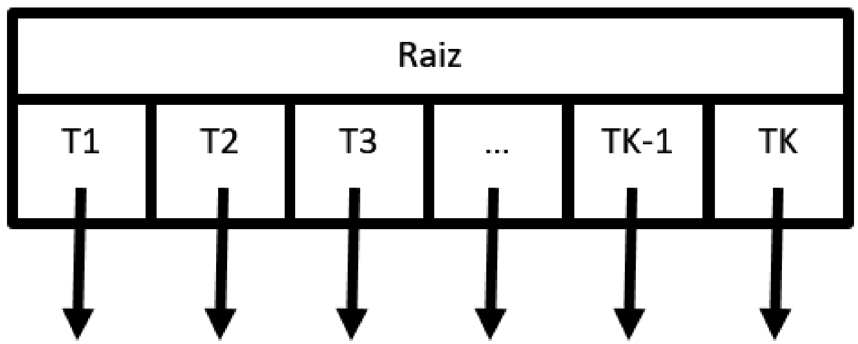
PROXHERMANO(CONSTRUIR(i,a,b)) $\equiv$ b

RAIZ(ARBOLVACIO) $\equiv$ indefinido

RAIZ(CONSTRUIR(i,a,b)) $\equiv$ i

Implementación ARBOL

- Una forma de **implementar** un árbol es tener en cada nodo además de sus datos, un puntero a cada hijo del nodo.

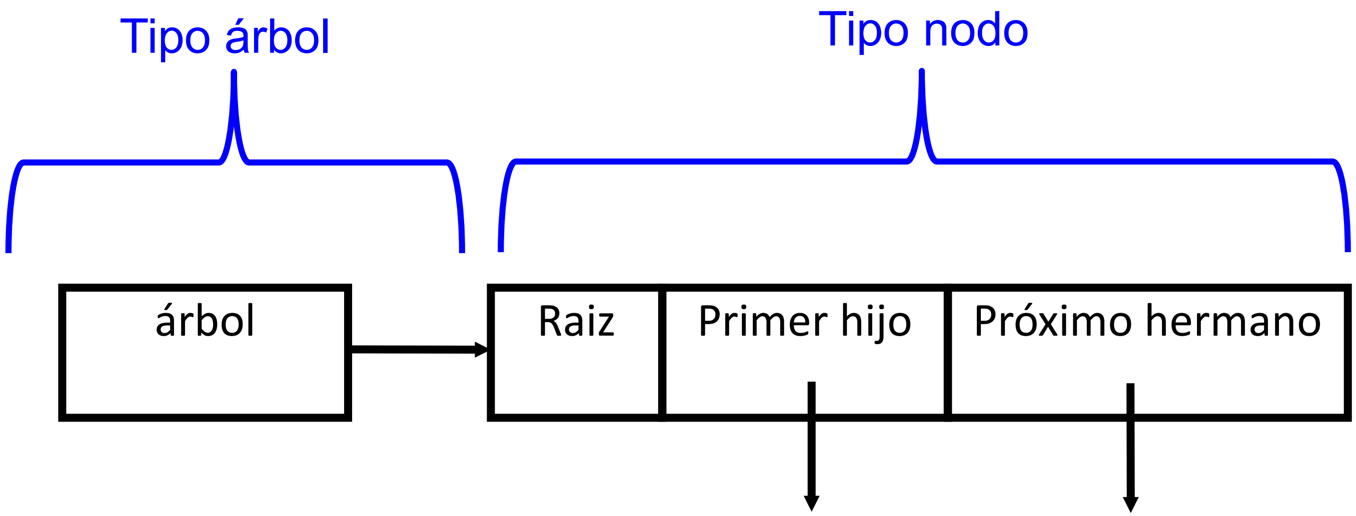


- Sin embargo como la cantidad de hijos por nodo puede variar mucho, y no se conocen de antemano, será un gasto innecesario de memoria un arreglo de punteros a los hijos.

Implementación ARBOL

La solución mas eficiente en memoria: cada nodo contiene además de la información, un puntero al primer hijo y un puntero al próximo hermano.

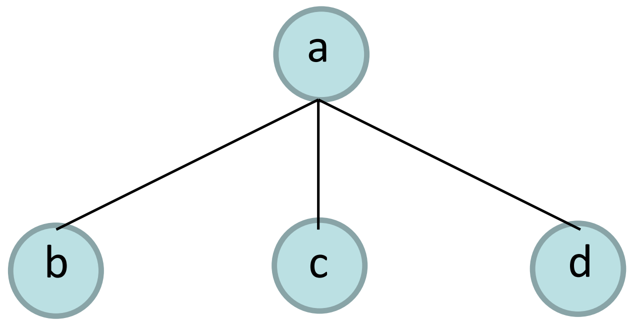
El tipo árbol será un puntero a un nodo.



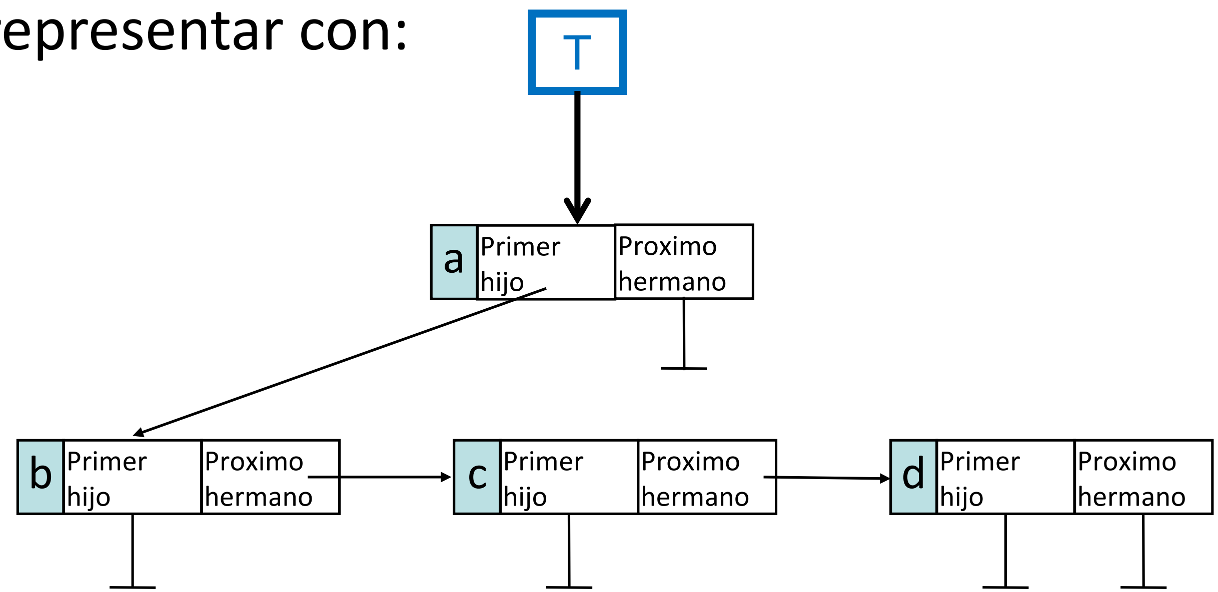
Implementación ARBOL

Ejemplo:

El arbol T:

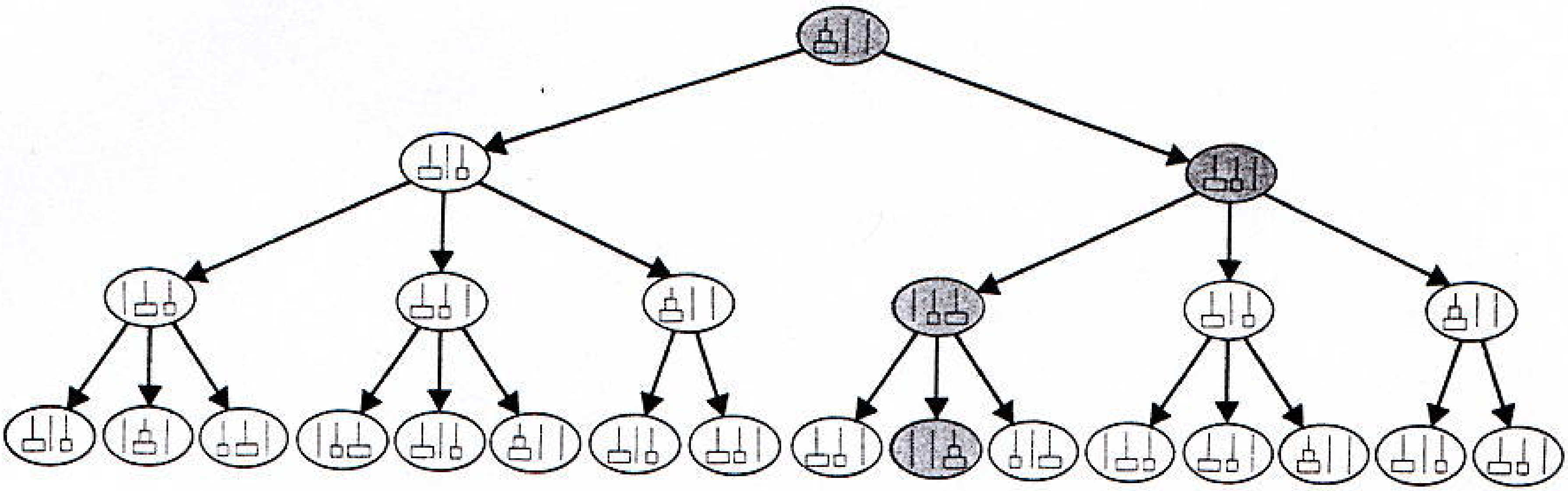


Se puede representar con:



APLICACIÓN:
Arbol de Juego

Torres de Hanoi con dos discos:



APLICACIÓN:
Arbol de Juego

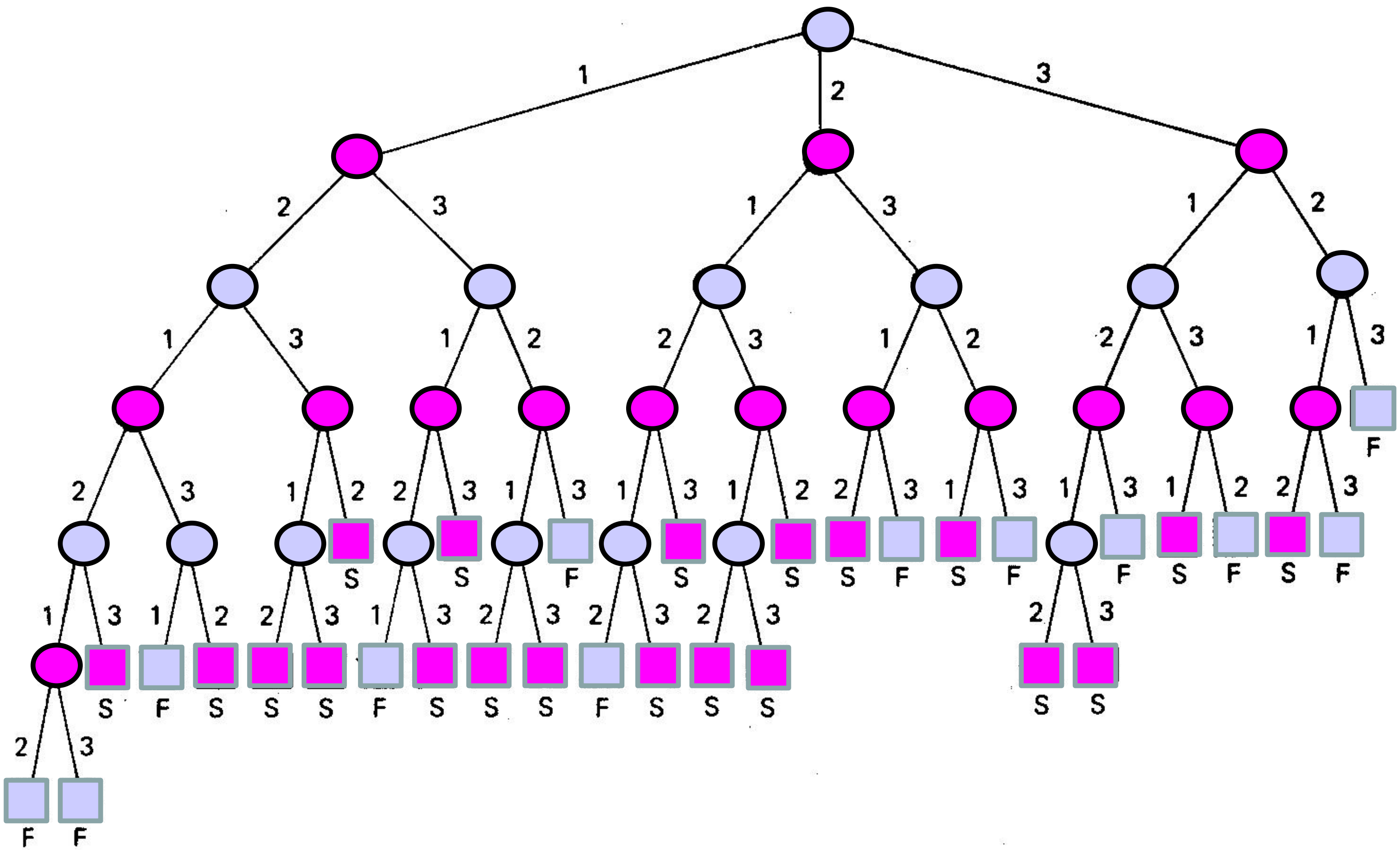
En el **Juego del ocho** participan 2 jugadores en jugadas alternadas, el primer jugador elige uno de los siguientes números: 1, 2 ó 3.

En cada turno posterior el jugador que le toque el turno, elige entre 1, 2 ó 3, pero el seleccionado previamente no puede elegirse.

Se elige una sucesión de números y si un jugador logra que la suma de los ya elegidos sea exactamente 8 habrá ganado. Si se pasa de 8 gana el otro jugador.

- Se puede observar que a pesar de ser un juego trivial, el árbol de juego tiene una dimensión considerable.

APLICACIÓN:
Juego del ocho



APLICACIÓN:
Arbol de decisión

Problema de las 8 monedas:

Dadas las 8 monedas a,b,c,d,e,f,g,h.

Se sabe que una de ellas es falsa y tiene diferente peso que las otras.

Se quiere determinar cual es la falsa usando una balanza de dos brazos.

Resolverlo usando el mínimo número de comparaciones y al mismo tiempo determinar si la moneda falsa es más liviana o más pesada que las demás.

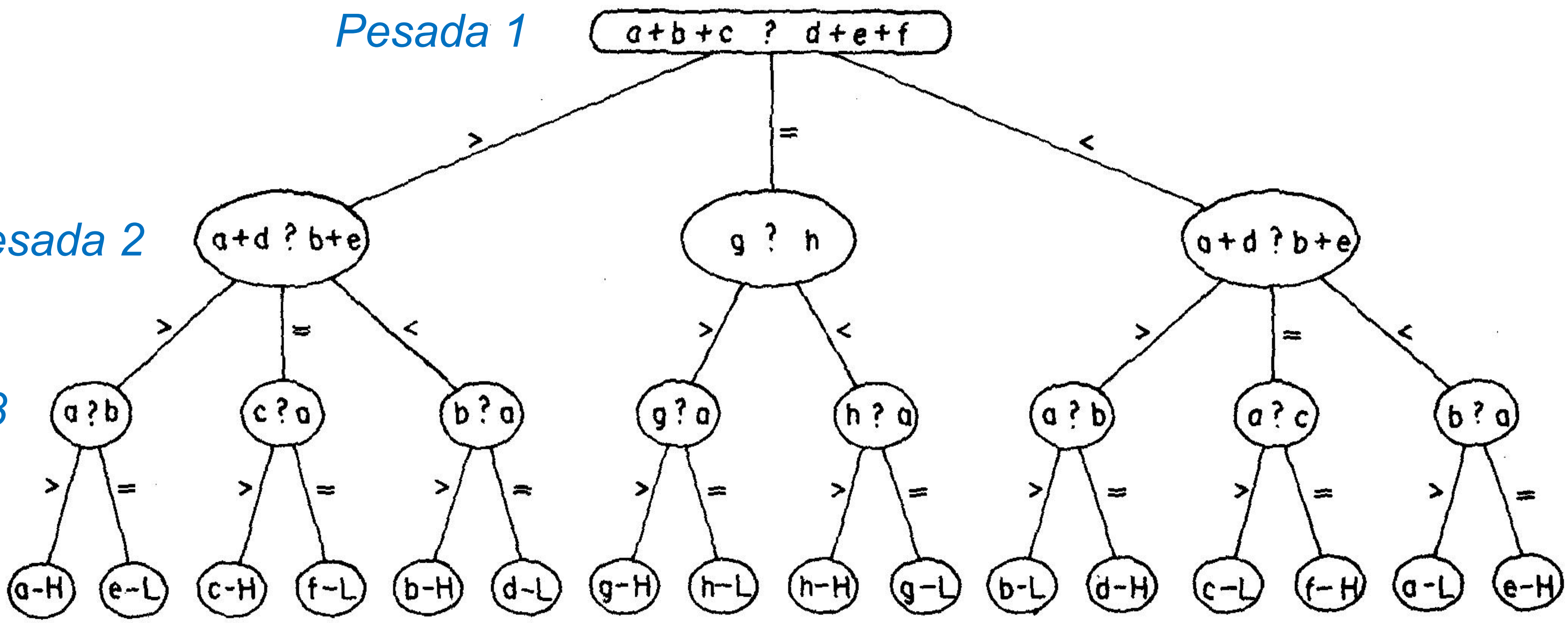
APLICACIÓN

ARBOL DE DECISIÓN - Problema de las 8 monedas

Pesada 1

Pesada 2

Pesada 3



<u>APLICACIÓN:</u>
Arbol de decisión

Problema de las 8 monedas:

El árbol representa el conjunto de decisiones por el cual se puede encontrar una respuesta al problema. Se llama árbol de decisión.

En el árbol se cubren todas las probabilidades, hay 16 nodos terminales.

Cada hoja tiene además el rótulo de la moneda más el hecho de ser más pesada (H) o más liviana (L).

Cada camino requiere 3 comparaciones.