

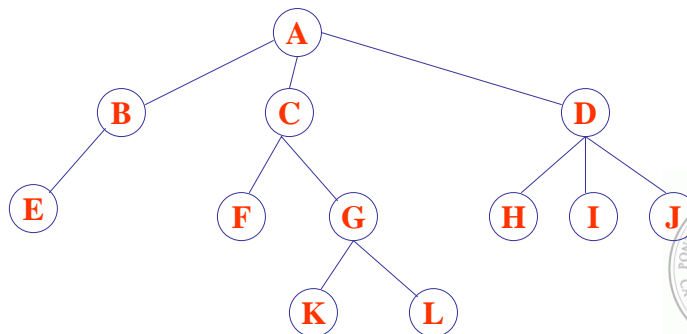
# ÁRBOLES

Prof. Nibaldo Rodriguez A.



## ÁRBOLES

- Un árbol A es un conjunto finito de uno o más nodos tal que:
  1. Existe un nodo especial denominado **RAIZ**( $V_1$ ) del árbol.
  2. Los nodos restantes ( $V_2, \dots, V_n$ ) se dividen en  $m \geq 0$  conjuntos disjuntos denominados  $A_1, A_2, A_3, \dots, A_m$ , cada uno de los cuales es, a su vez, un árbol. Estos árboles se llaman **Subárboles de la RAIZ**.





## ÁRBOLES

**Raíz:** Nodo que no tiene antecesor

**Nodo:** Vértices o elementos del árbol

**Nodo Terminal u hoja:** Vértices o elementos del árbol que no contienen subárboles.

**Hermanos:** Nodos de un mismo padre.

**Nodos Interiores:** Nodos que no son hoja ni raíz.

**Bosque:** Colección de dos o más árboles.

**Arista:** Enlace entre dos nodos consecutivos.

**Camino:** Secuencia de aristas consecutivas.

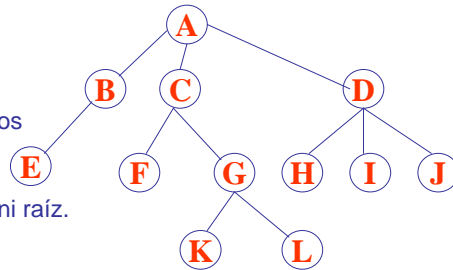
**Rama:** Camino que termina en hoja.

**Nivel:** longitud del camino desde la raíz al nodo específico.

**Altura o profundidad:** el número máximo de nodos de una rama. Equivale al nivel más alto de los nodos más uno.

**Peso:** es el número de nodos terminales.

**Grado:** el número de hijos que salen de un nodo

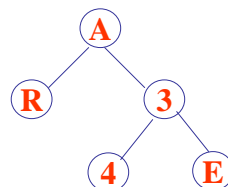
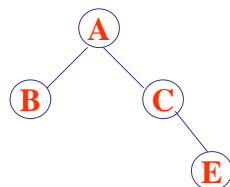


## Arboles Binarios

● Un Árbol binario es un conjunto finito de cero o más nodos tal que:

1. Existe un nodo denominado raíz del árbol
2. Cada nodo tiene 0, 1 o 2 subárboles,
  - Llamado Subárbol Izquierdo y Subárbol Derecho.

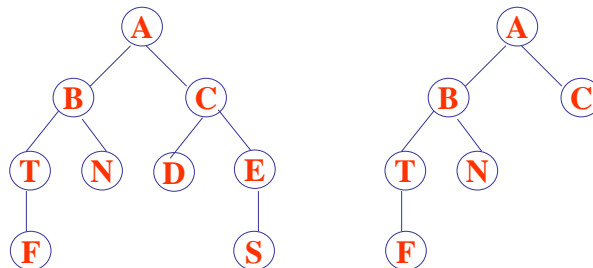
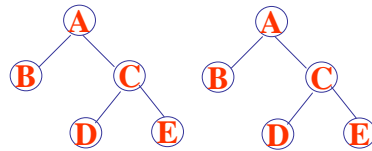
● Dos Árboles Binarios son **Similares** si tienen la misma estructura





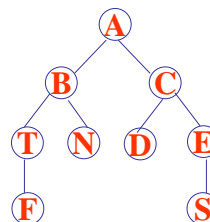
## Arboles Binarios

- Se dice que dos Árboles Binarios son **equivalentes** si tienen la misma estructura y además la misma información
- Se dice que un Árbol Binario es **EQUILIBRADO** si las alturas de los dos Subárboles de cada nodo del árbol se diferencian en una unidad como máximo.

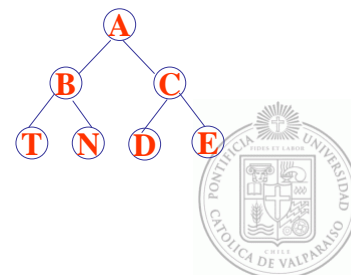


## Arboles Binarios

- Un árbol binario es **Completo** si todos sus nodos, excepto las hojas, tienen exactamente dos subárboles.



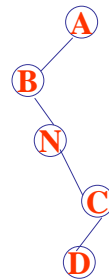
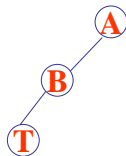
- Un árbol binario es **lleno** si todas sus hojas están al mismo nivel y todos sus nodos interiores tienen cada uno 2 hijos.





## Arboles Binarios

- Un árbol binario es degenerado si todos sus nodos excepto el último tienen sólo un subárbol



## PROPIEDAD

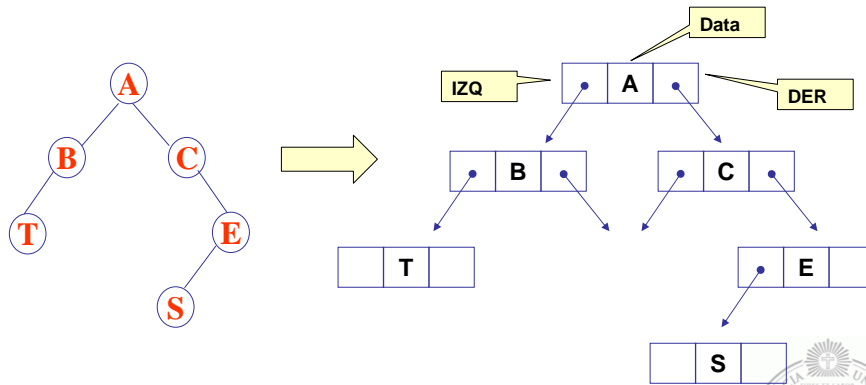
- Dado un árbol de grado  $g$  y altura  $h$ , el número máximo de nodos es igual a:

$$N_{nodo} = \sum_{i=0}^{h-1} g^i$$

- Determinar el número de nodos para el caso  $g=2$  (árbol binarios)



## Estructuras de Datos para Arboles Binarios



## DCLARACIÓN EN LENGUAJE C

```
typedef struct nodo {  
    int data;  
    struct nodo *izq, *der;  
} Arbol;
```





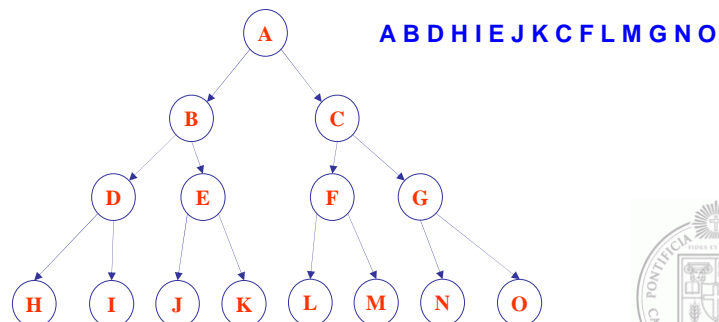
## Recorrido de árboles binarios

- Se denomina recorrido de un árbol al proceso que permite acceder una sola vez a cada uno de los nodos del árbol. Cuando un árbol se recorre, el conjunto completo de nodos se examina.
- **Recorrido pre-orden**
  1. Visitar nodo raíz
  2. Recorrer el subárbol izquierdo en modo pre-orden
  3. Recorrer el subárbol derecho en modo pre-orden
- **Recorrido in-orden**
  1. Recorrer el subárbol izquierdo en modo in-orden
  2. Visitar nodo raíz
  3. Recorrer el subárbol derecho en modo in-orden
- **Recorrido post-orden**
  1. Recorrer el subárbol izquierdo en modo post-orden
  2. Recorrer el subárbol derecho en modo post-orden
  3. Visitar nodo raíz



## Recorrido Pre-Orden

- **Recorrido Pre-Orden**
  1. Visitar nodo raíz
  2. Recorrer el subárbol izquierdo en modo pre-orden
  3. Recorrer el subárbol derecho en modo pre-orden

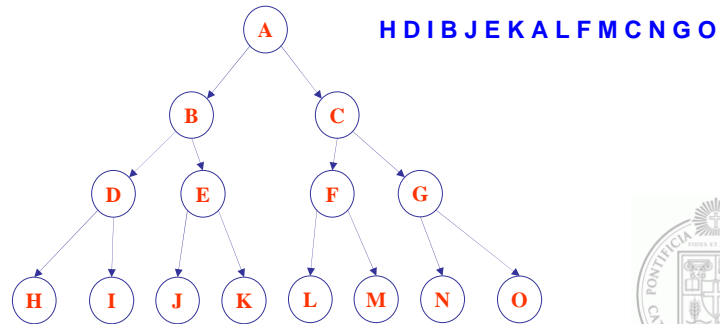




### Recorrido in-orden

#### ● Recorrido In-Orden

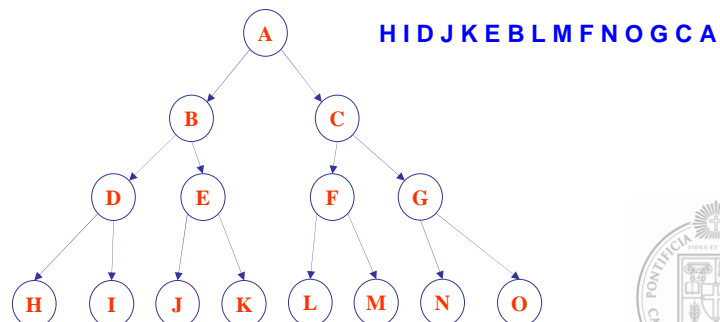
1. Recorrer el subárbol izquierdo en modo in-orden
2. Visitar nodo raíz
3. Recorrer el subárbol derecho en modo in-orden



### Recorrido post-orden

#### ● Recorrido Post-Orden

1. Recorrer el subárbol izquierdo en modo post-orden
2. Recorrer el subárbol derecho en modo post-orden
3. Visitar nodo raíz



# EJERCICIOS

- Dado un árbol binario de datos entero. Implementar las siguientes funciones iterativas:
- Recorrido:
  - Preorden, Inorden, Postorden
- Recorrido:
  - Anchura Izquierdo-derecho
  - Anchura Derecho-Izquierdo



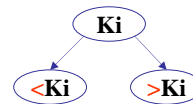
## Arboles binarios de búsqueda

- Un árbol de búsqueda es un TDA árbol en el que para cada nodo todas las claves de cada subárbol satisfacen una y solo una condición de un conjunto de Condiciones mutuamente excluyentes
- **Un árbol binario de búsqueda, es un árbol binario en el que dadas dos Condiciones mutuamente excluyentes, para cada nodo, todas las claves de su subárbol izquierdo satisfacen una condición y todas las de su subárbol Derecho la otra.**

### EJEMPLO DE NÚMEROS ENTEROS:

Para cada nodo  $N_i$  con clave  $K_i$ , todas las claves en los nodos del subárbol izquierdo son menores que  $K_i$  y todas las claves en los nodos del subárbol derecho son mayores que  $K_i$ .

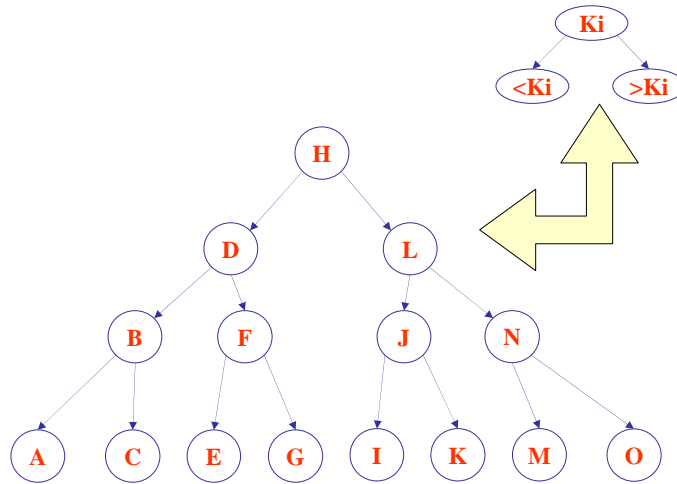
Siempre existen 2 relaciones mutuamente excluyentes ( $<$  ;  $>$ )







## Arboles binarios de búsqueda



## DECLARACIÓN EN LENGUAJE C

```
typedef struct nodo {  
    int dato;  
    struct nodo *izq,*der;  
} Arbol;
```



## IMPLEMENTACIÓN ITERATIVA ABB

- INSERTAR
- ELIMINAR DATO usando criterio del menor de los mayores