

# UNIDAD I. ESTRUCTURA DE DATOS BÁSICAS (ÁRBOLES)

Francisco J. Hernández López

fcoj23@cimat.mx



# ÁRBOLES

- Estructura de datos no lineal, en la que cada elemento sólo puede estar enlazado con su predecesor (o nodo padre) y sus sucesores (o nodos hijos)
- Existe un único camino entre el primer nodo de la estructura y cualquier otro nodo
- Se utilizan para representar jerarquías:
  - Árbol genealógico
  - Diagramas de organización
  - Formulas matemáticas
  - Numerar capítulos y secciones de un libro
  - Algoritmos para ordenación, búsqueda, compilación, etc.

# DEFINICIONES

- **Nodo:** Vértices o elementos de un árbol
- **Enlace/arco/borde/arista:** Conexión entre dos nodos consecutivos
- **Nodo:**
  - **Raíz:** Todo árbol que no es vacío, tiene un único nodo raíz, el cual es el nodo superior de la jerarquía
  - **Terminal u hoja:** Nodo que no contiene ningún subárbol, o que no tiene hijos
  - **Interior o rama:** Nodo con uno o más subárboles, o nodo que no es hoja
  - **Descendiente o hijo:** Cada subárbol de un nodo
  - **Ascendiente o padre:** Nodo de jerarquía superior a un nodo dado
  - **Hermanos:** Nodos que tienen el mismo padre

# DEFINICIONES

- **Bosque:** Colección de árboles
- **Orden del árbol:** Es el número de hijos que puede llegar a tener cada elemento del árbol. Ej.: En un árbol binario, cada nodo solo puede llegar a tener dos hijos, por lo tanto el orden del árbol es 2
- **Grado del árbol:** Es el número de hijos que tiene el elemento con más hijos dentro del árbol
- **Nivel de un nodo:** Es la distancia (medida en nodos) que hay entre el nodo y la raíz. Si consideramos que el nivel de la raíz es 1, entonces el nivel que tienen sus hijos es 2

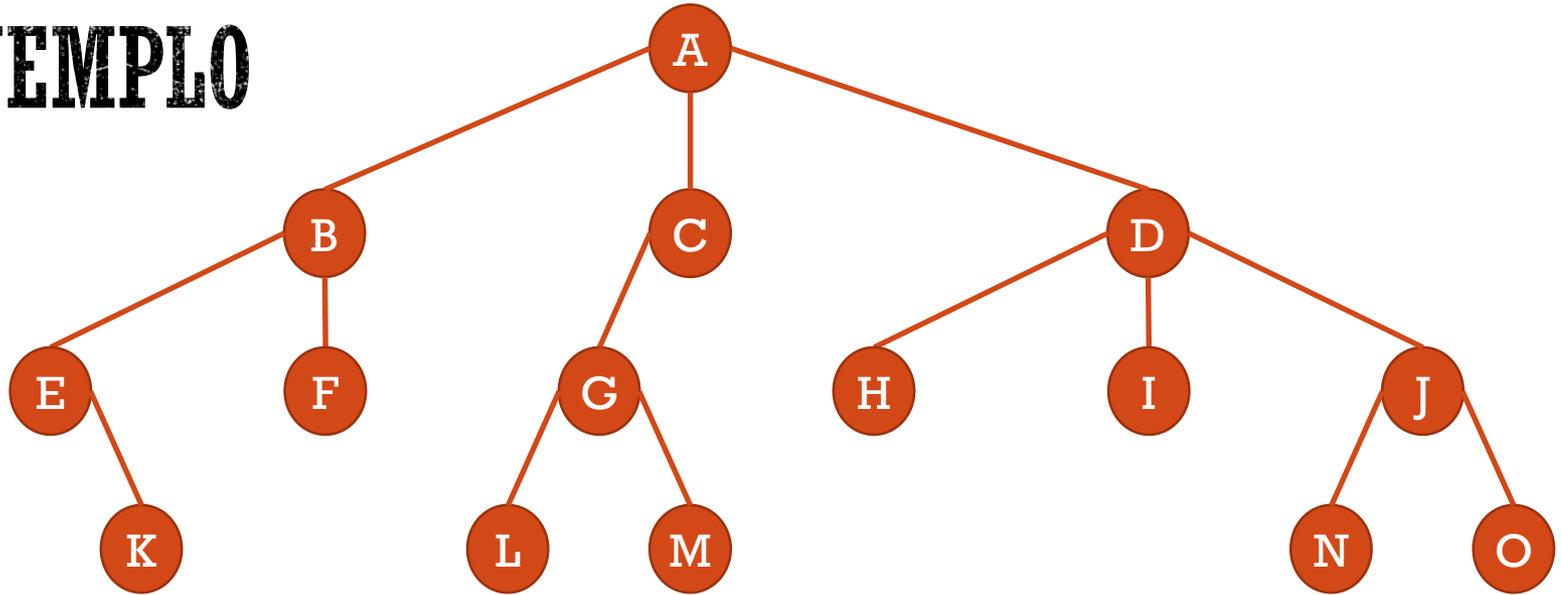
Nota: También se puede considerar que la raíz tiene nivel 0. Entonces el nivel de un nodo será la longitud del camino (medido en enlaces) desde la raíz al nodo

# DEFINICIONES

- **Altura del árbol:** Es el nivel del nodo de mayor nivel. Como cada nodo del árbol se puede considerar a su vez como la raíz de un árbol, entonces podemos hablar de altura de ramas
- **Peso del árbol:** Número de nodos terminales (u hojas)
- **Factor de equilibrio:** Se define como la altura del subárbol derecho menos la altura del subárbol izquierdo

$$FE = A_{SD} - A_{SI}$$

# EJEMPLO



Raíz: A

Hojas: F, H, I, K, L, M, N, O

Ramas: B, C, D, E, G, J

Hijos: B, C, D, E, F, G, H, I, J, K, L, M, N, O

Padres: A, B, C, D, E, G, J

Hermanos: (B, C, D) de A, (E, F) de B,  
(H, I, J) de D, (L, M) de G, (N, O) de J

Orden del árbol: 3

Grado del árbol: 3

Nivel del nodo F: 3

Altura del árbol: 4

Altura de la rama B: 3

Altura de la rama I: 1

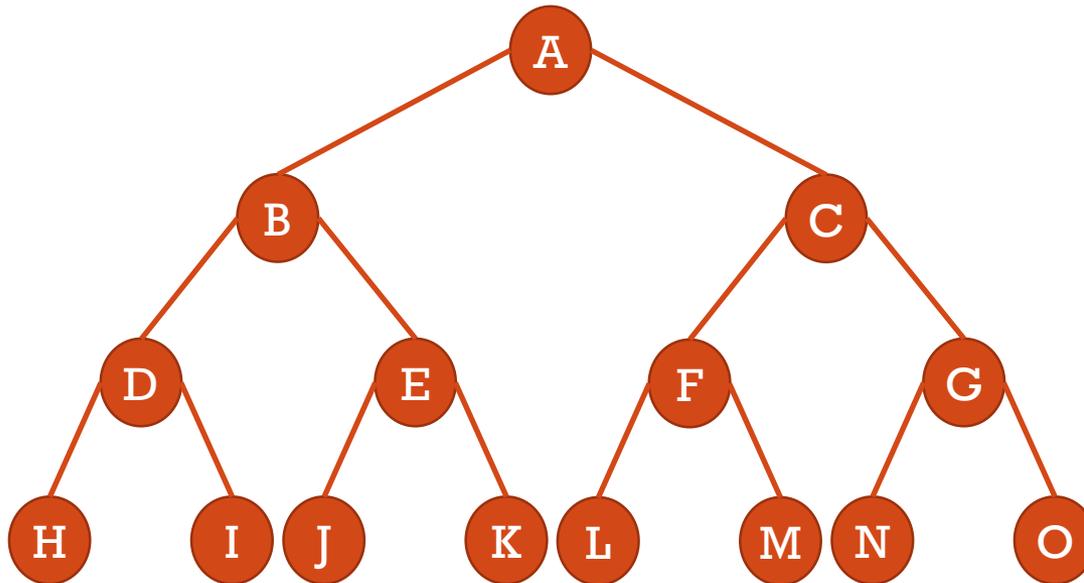
Peso del árbol: 8

*FE* en el nodo D: 1 (Sin considerar el  
nodo I)

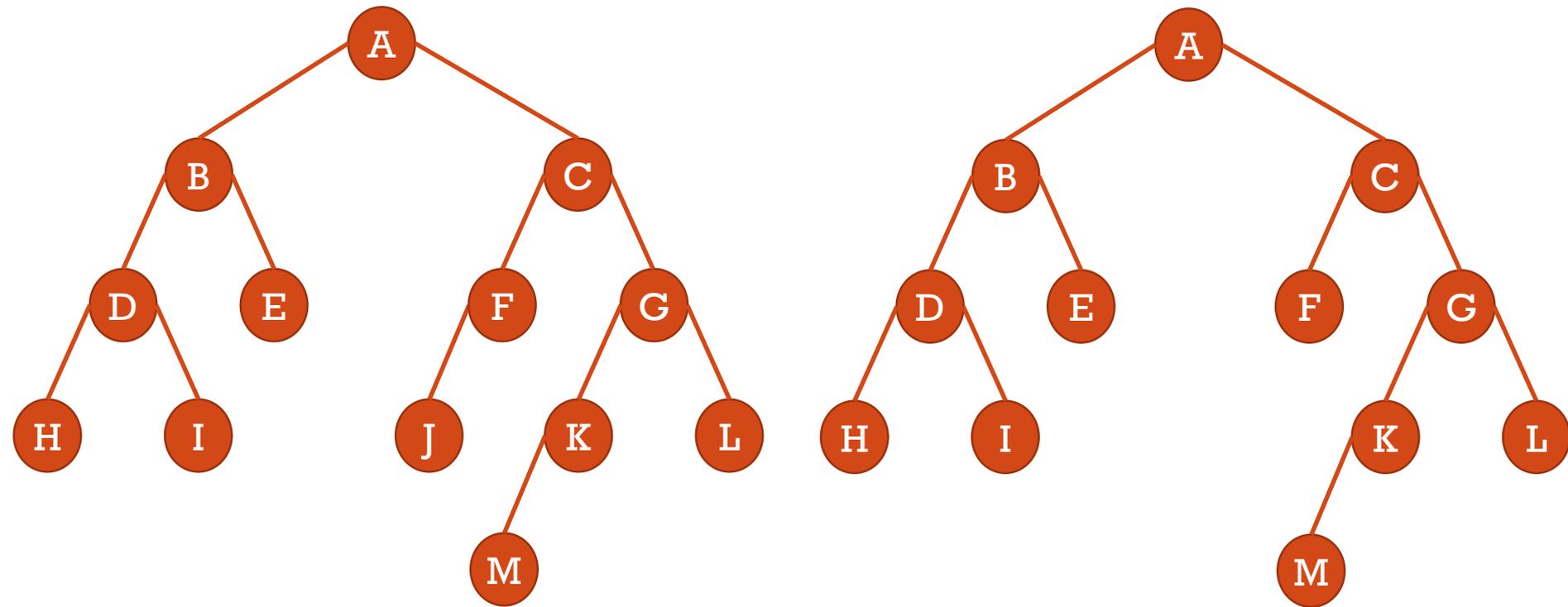
# ÁRBOLES BINARIOS

- Son árboles en los que cada nodo no puede tener más de dos hijos, por lo tanto son árboles de orden 2
- Tipos de árboles binarios:
  - **Completamente balanceados/Completos/Perfectos:** Si cada nodo tiene exactamente dos hijos o no tiene hijos y si cada hoja está al mismo nivel. Un árbol binario de nivel  $n$  tiene  $2^n - 1$  nodos
  - **Equilibrados:** Las alturas de los dos subárboles de cada nodo tiene como máximo una diferencia de 1 en valor absoluto, es decir el  $|FE| \leq 1$  en cada nodo
  - **Degenerados:** Todos sus nodos solo tienen un subárbol
  - **Similares:** Árboles con la misma estructura
  - **Equivalentes:** Árboles con la misma estructura y contienen la misma información

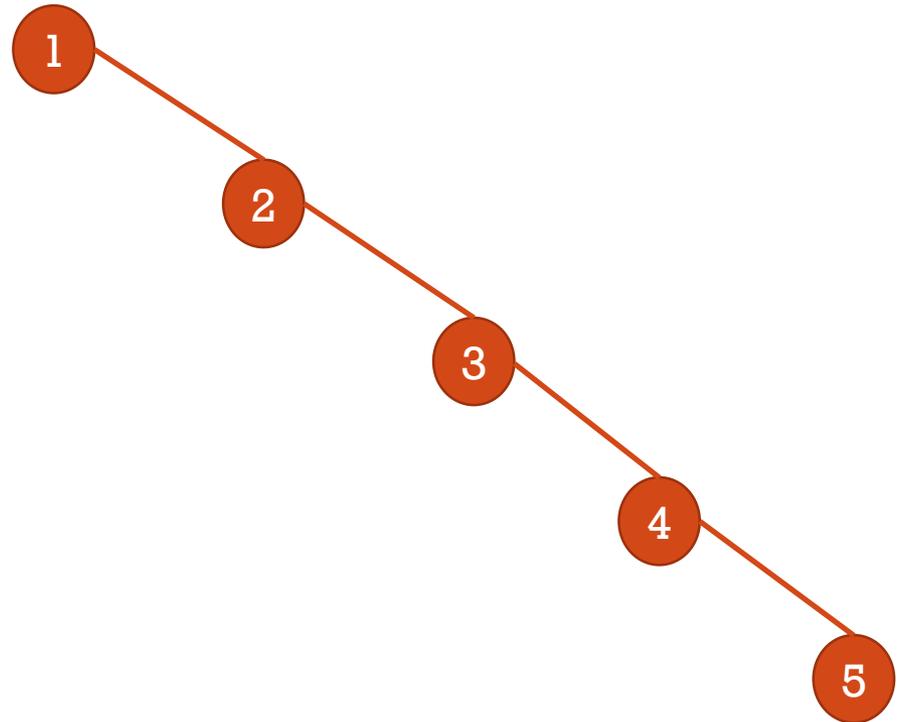
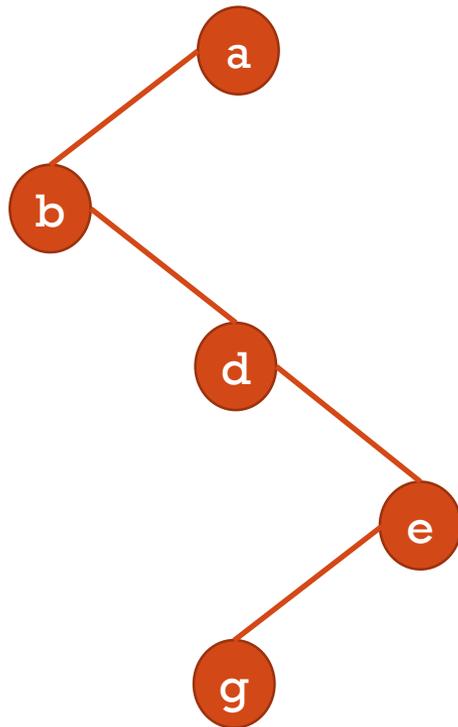
# ÁRBOLES COMPLETOS



# EQUILIBRADO Y NO EQUILIBRADO

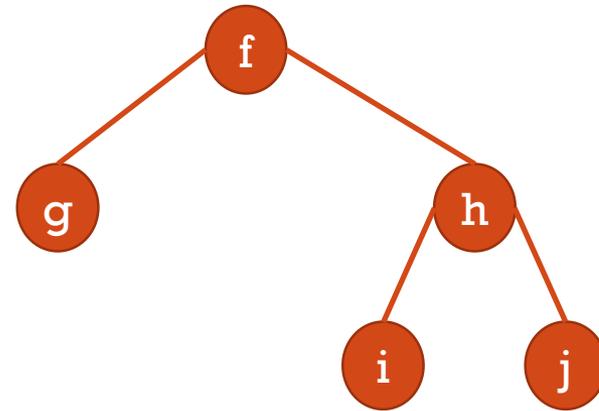
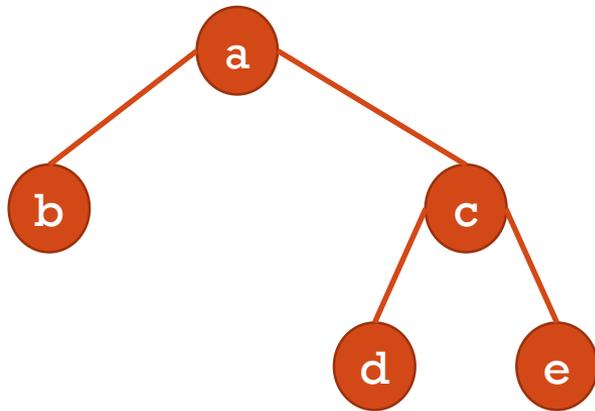


# ÁRBOLES DEGENERADOS

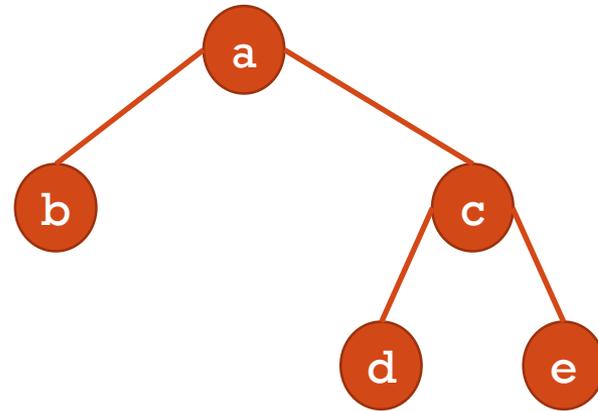
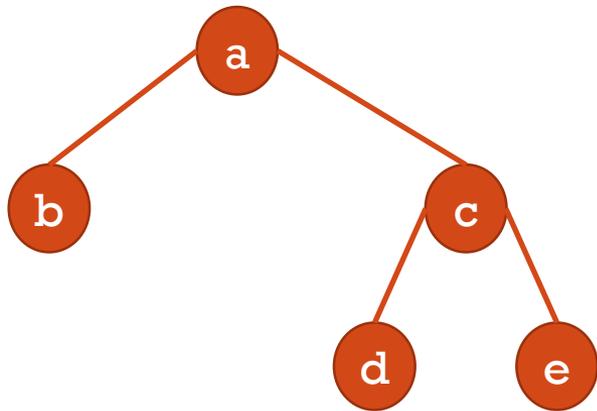


Nota: Estos árboles tienen orden 2 y grado 1

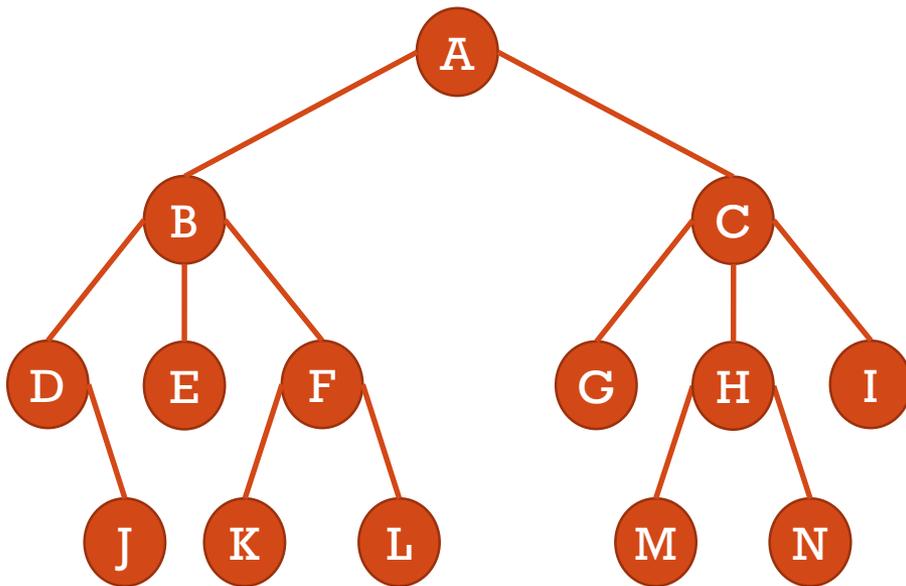
# ÁRBOLES SIMILARES



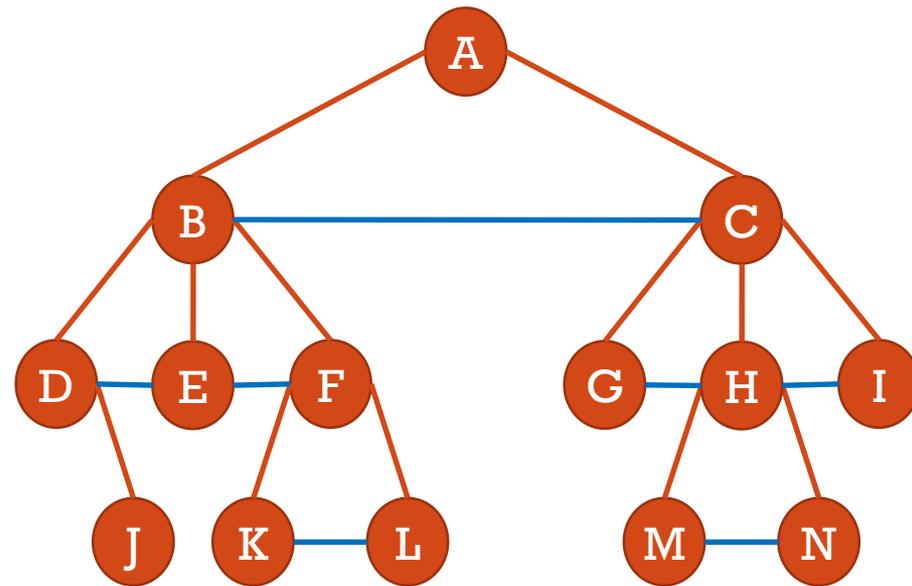
# ÁRBOLES EQUIVALENTES



# CONVERTIR UN ÁRBOL GENERAL EN UN ÁRBOL BINARIO

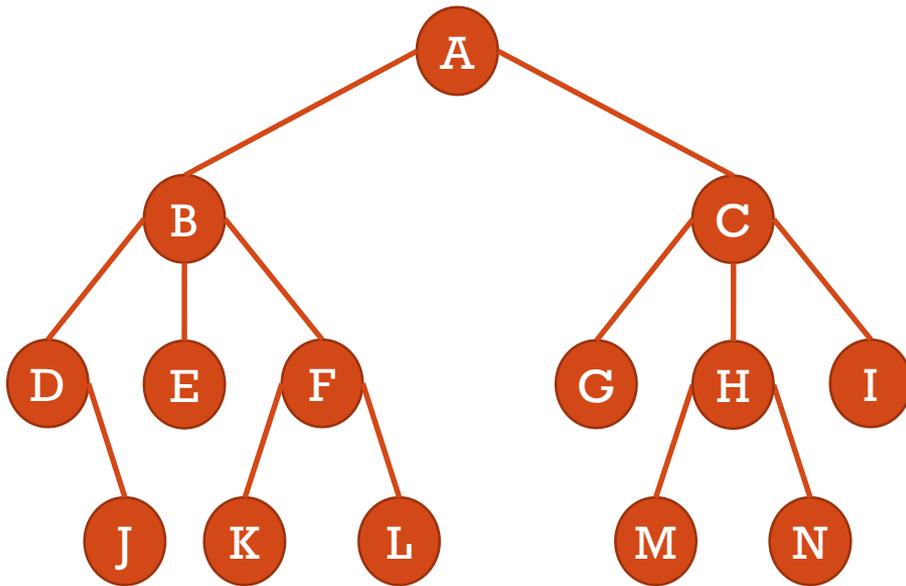


Árbol de orden 3

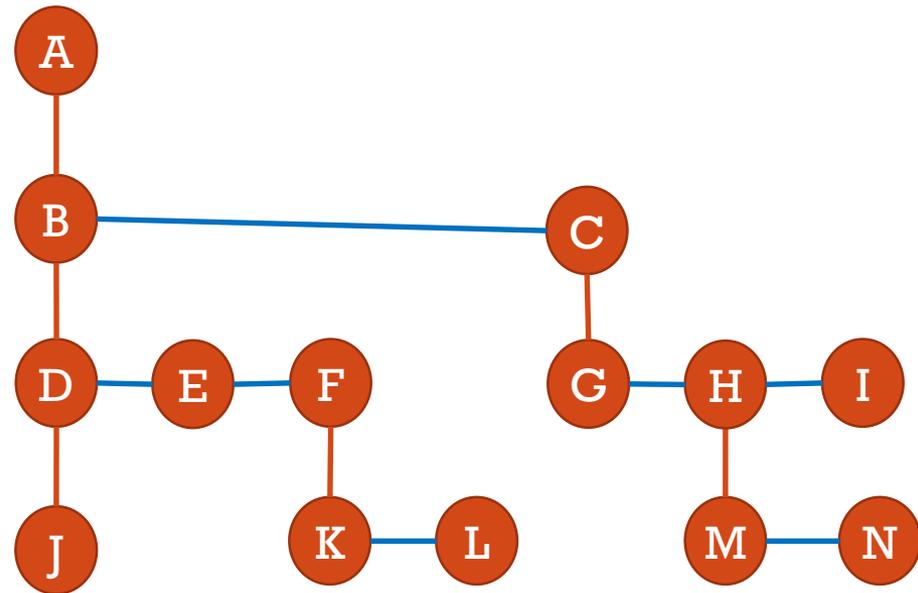


1. Enlazar los hijos de cada nodo en forma horizontal (todos los hermanos)

# CONVERTIR UN ÁRBOL GENERAL EN UN ÁRBOL BINARIO

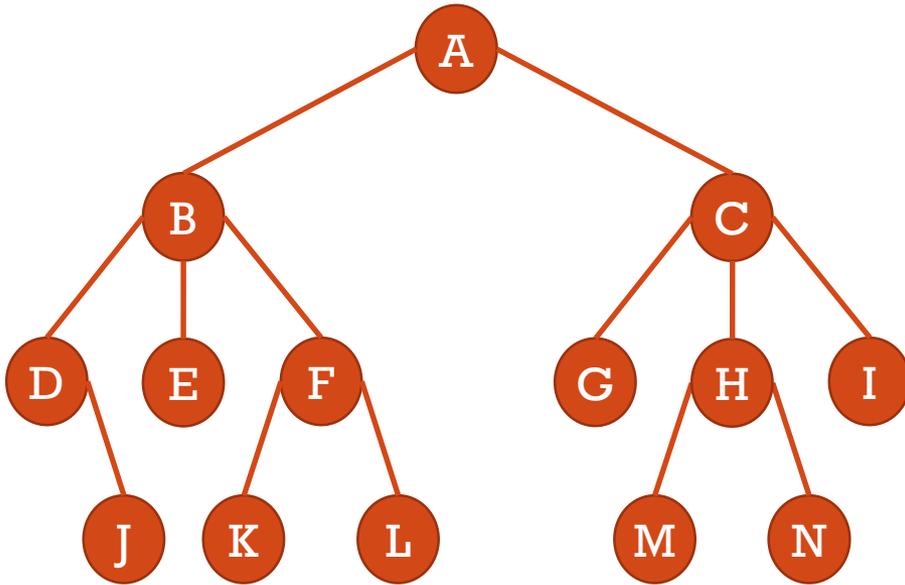


Árbol de orden 3

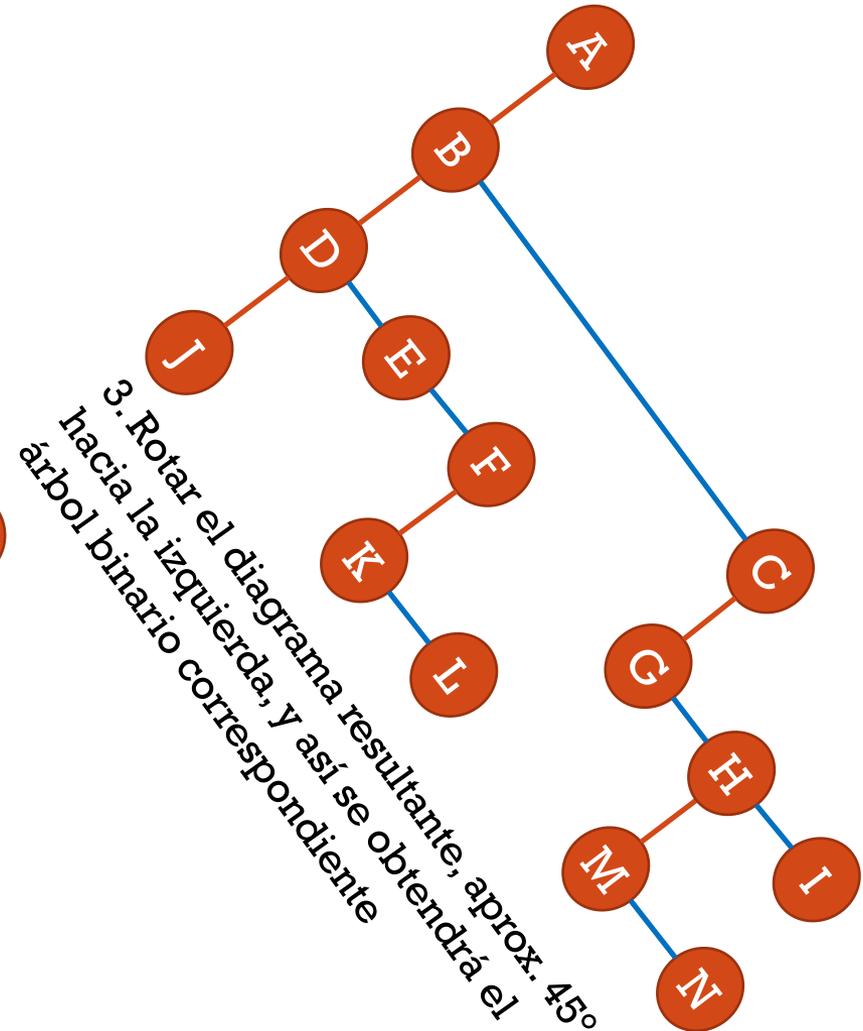


2. Enlazar en forma vertical el nodo padre con el hijo que se encuentra más a la izquierda. Además debe eliminarse el enlace de dicho padre con el resto de sus hijos

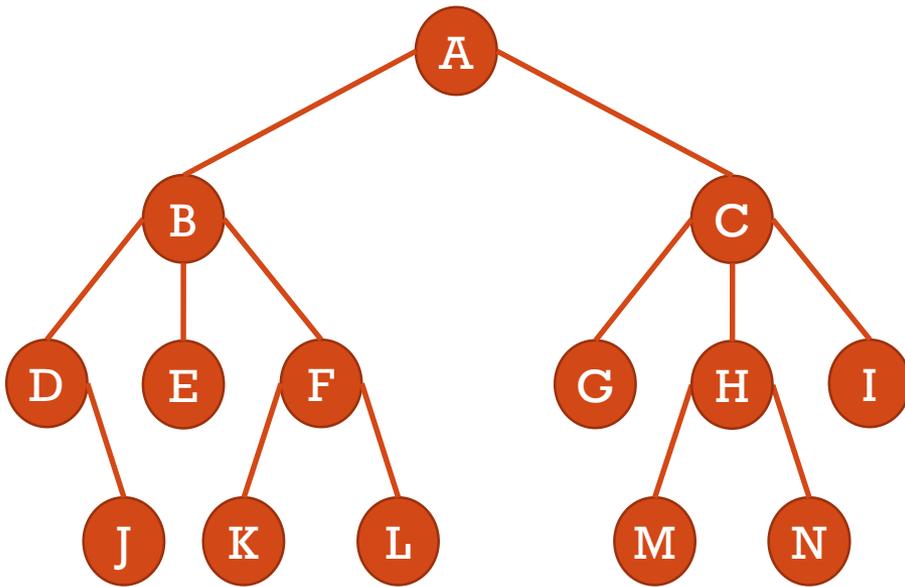
# CONVERTIR UN ÁRBOL GENERAL EN UN ÁRBOL BINARIO



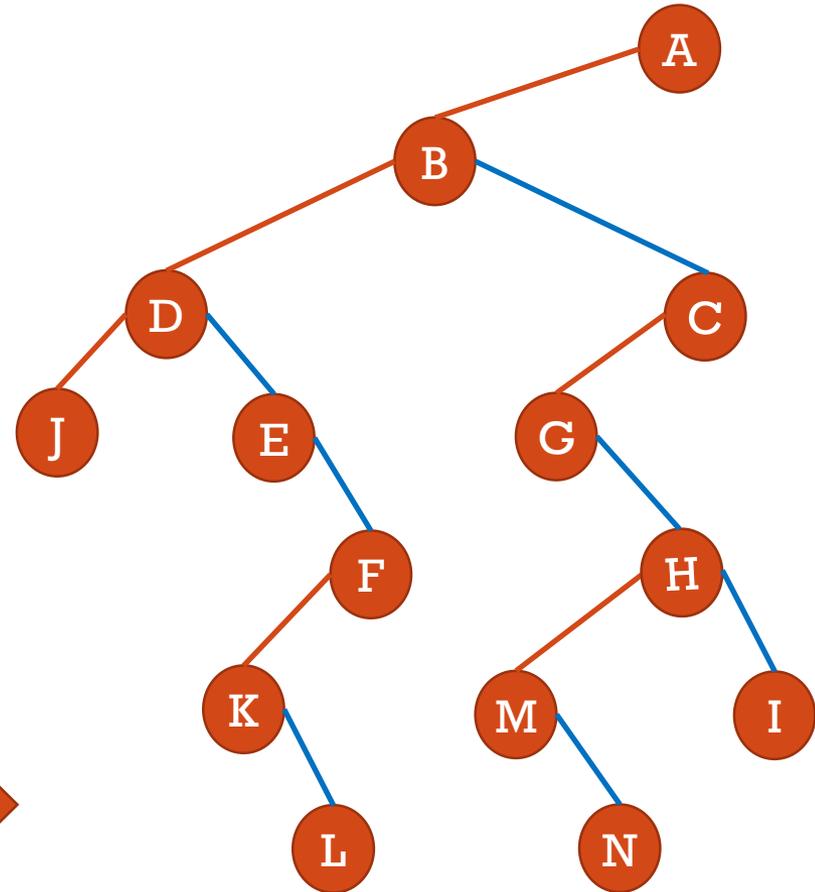
Árbol de orden 3



# CONVERTIR UN ÁRBOL GENERAL EN UN ÁRBOL BINARIO

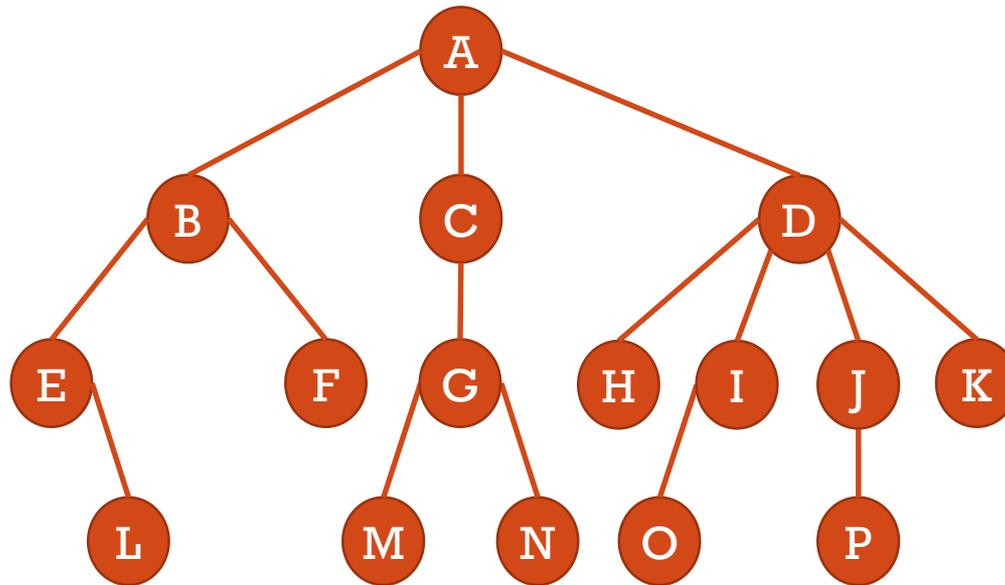


Árbol de orden 3

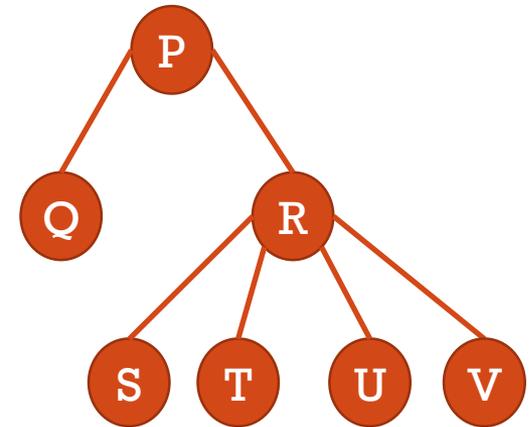
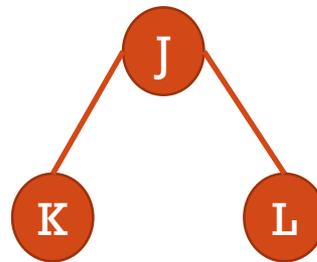
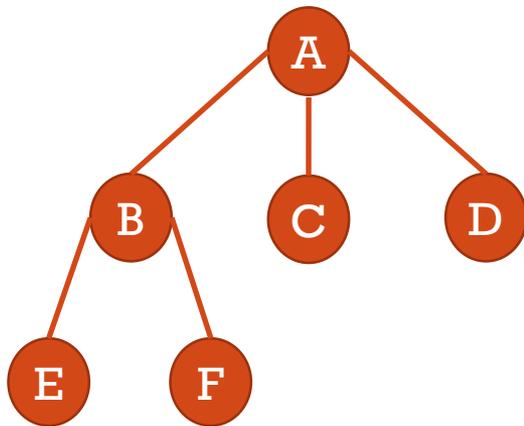


Árbol binario

# CONVERTIR EL SIGUIENTE ÁRBOL EN UN ÁRBOL BINARIO

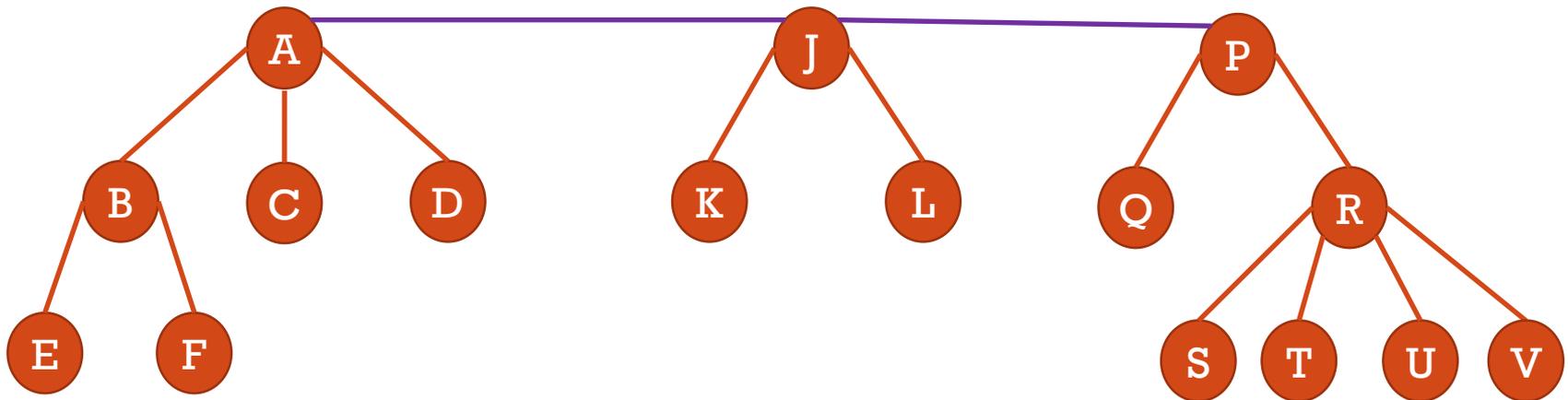


# CONVERTIR UN BOSQUE EN UN ÁRBOL BINARIO



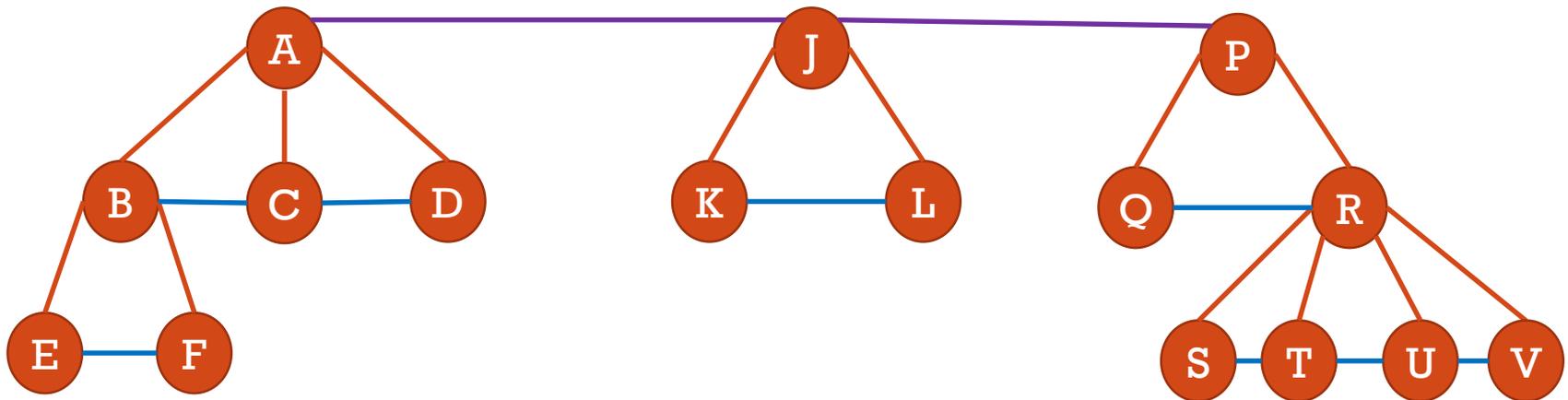
Bosque de árboles

# CONVERTIR UN BOSQUE EN UN ÁRBOL BINARIO



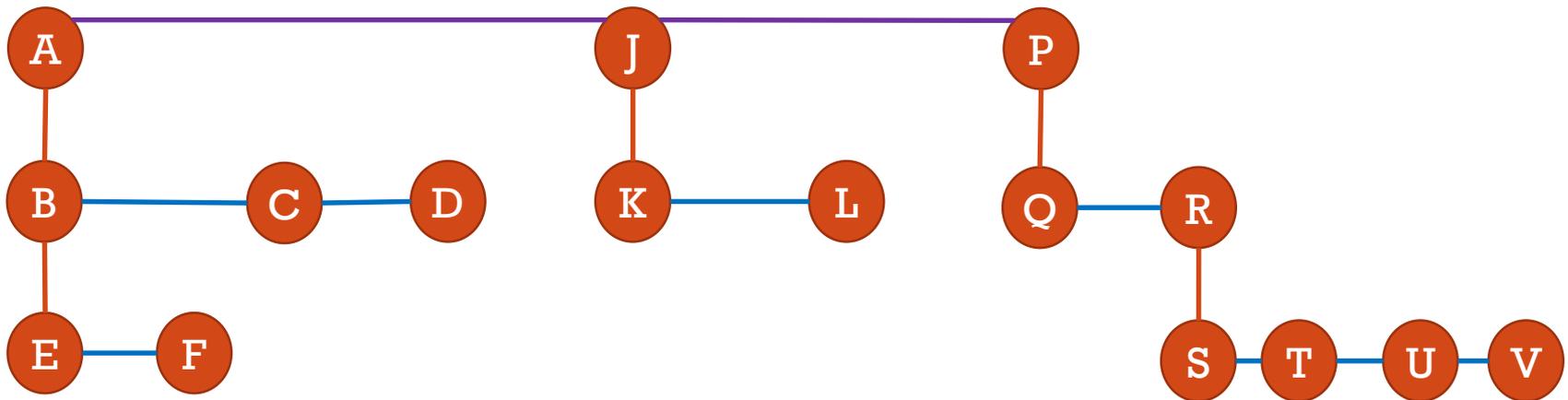
1. Enlazar en forma horizontal las raíces de los distintos árboles generales

# CONVERTIR UN BOSQUE EN UN ÁRBOL BINARIO



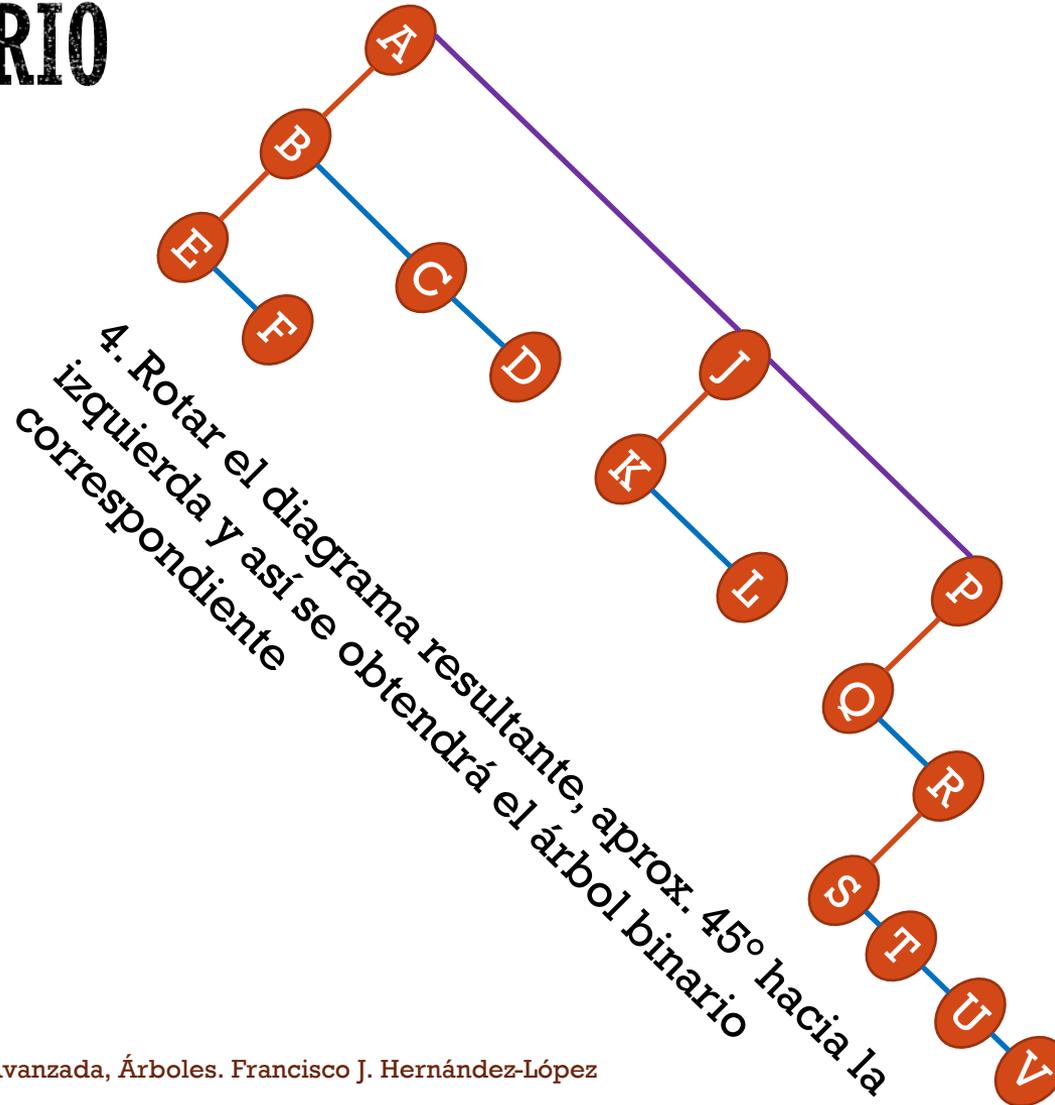
2. Enlazar los hijos de cada nodo en forma horizontal (todos los hermanos)

# CONVERTIR UN BOSQUE EN UN ÁRBOL BINARIO

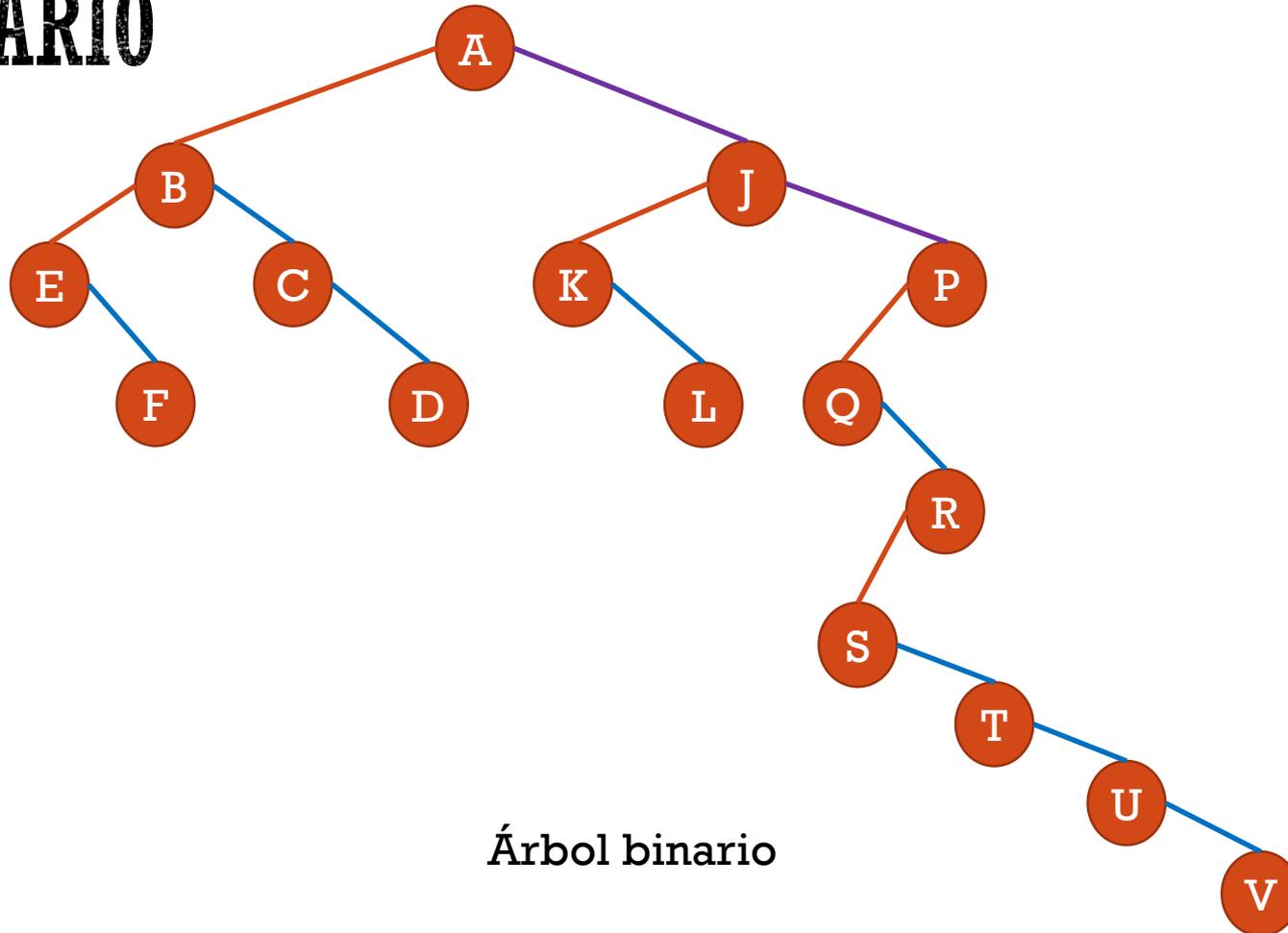


3. Enlazar en forma vertical el nodo padre con el hijo que se encuentra más a la izquierda. Además se debe eliminar el vínculo del padre con el resto de sus hijos

# CONVERTIR UN BOSQUE EN UN ÁRBOL BINARIO

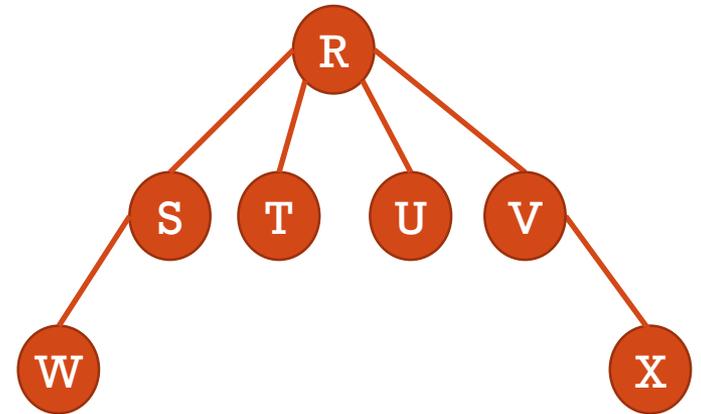
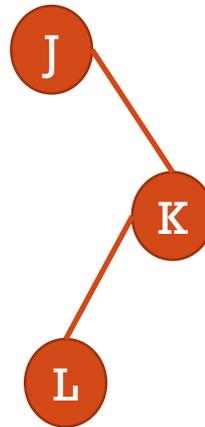
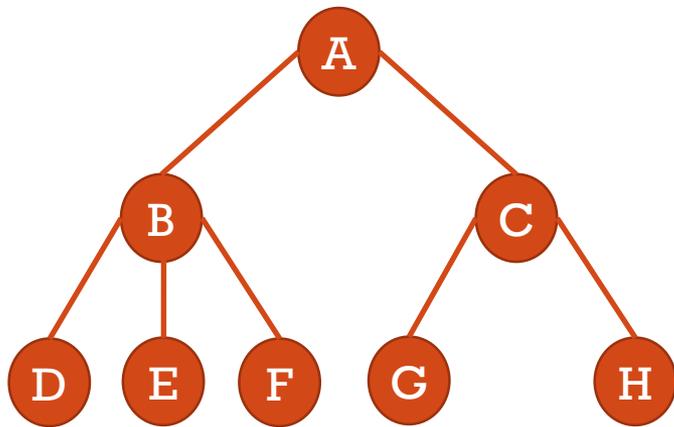


# CONVERTIR UN BOSQUE EN UN ÁRBOL BINARIO



Árbol binario

# CONVERTIR EL SIGUIENTE BOSQUE EN UN ÁRBOL BINARIO

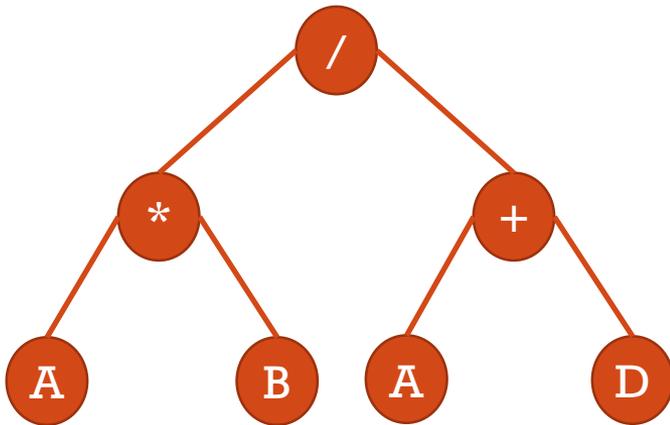


Bosque de árboles

# OPERACIONES BÁSICAS CON ÁRBOLES BINARIOS

- Añadir o insertar elementos
- Buscar o localizar elementos
- Borrar o eliminar elementos
- Moverse a través del árbol
- Recorrer todo el árbol

# FORMAS DE RECORRER TODO EL ÁRBOL BINARIO



Pre-orden:  $/*AB+AD$   
(raíz, izq, der)

In-orden:  $A*B/A+D$   
(izq, raíz, der)

Post-orden:  $AB*AD+ /$   
(izq, der, raíz)

# IMPLEMENTACIÓN DE UN ÁRBOL BINARIO

- Se pueden implementar tanto con memoria estática así como con memoria dinámica
- A continuación veremos como implementar un árbol binario con memoria dinámica:

```
typedef struct _nodoArbol{  
    int dato;  
    struct _nodoArbol *izq;  
    struct _nodoArbol *der;  
}tipoNodoArbol;
```