



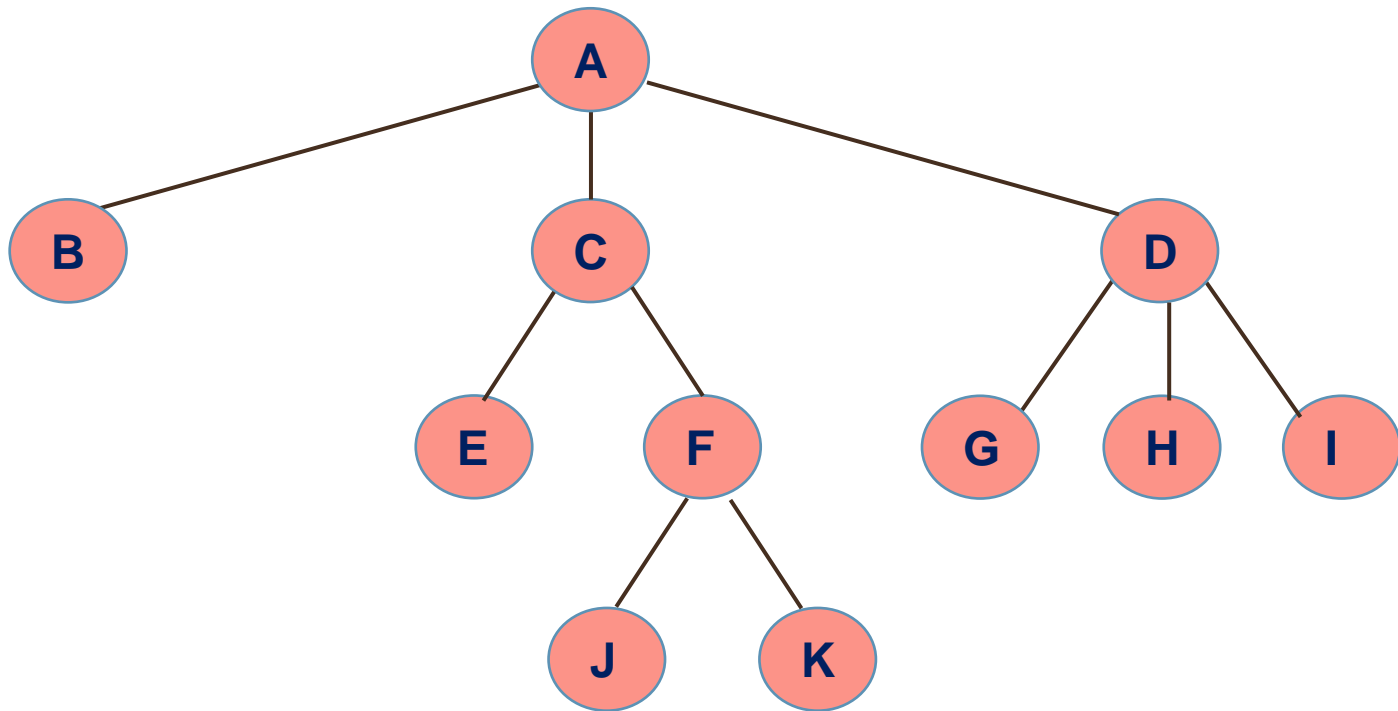
# Árboles Generales

**Ejercitación usando Recorrido  
por niveles**

# Ejercicios

- 1) ¿Cuántos **niveles** tiene el árbol?
- 2) ¿Cuántos **nodos** hay en **cada** nivel del árbol?
- 3) ¿Cuántos **nodos** hay en el nivel  $k$  del árbol?

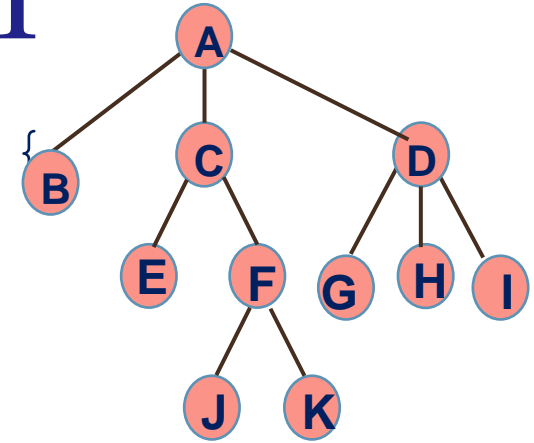
# Resolución de los Ejercicios



# Resolución del Ejercicio 1

## Seudocódigo Recorrido-Niveles

```
q: cola de vértices;  
encolar raíz R en q;  
mientras (cola no se vacíe) {  
    desencolar v de q;  
    imprimir (dato de v);  
    para cada hijo w de v  
        encolar w en q; }  
}
```



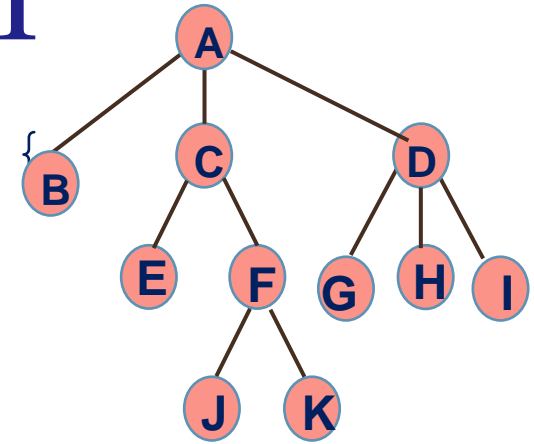
**A**

← Cola

# Resolución del Ejercicio 1

## Seudocódigo Recorrido-Niveles

```
q: cola de vértices;  
encolar raíz R en q;  
mientras (cola no se vacíe) {  
    desencolar v de q;  
    imprimir (dato de v);  
    para cada hijo w de v  
        encolar w en q; }  
}
```



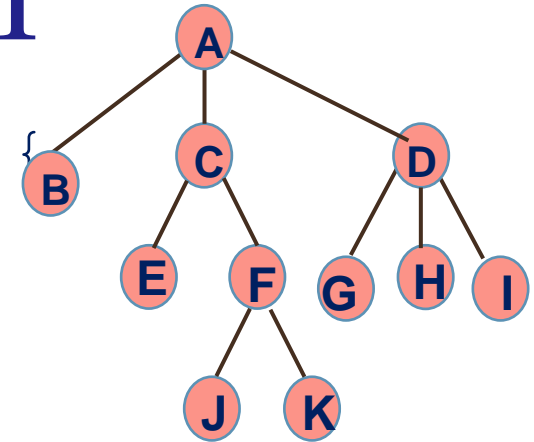
← Cola

# Resolución del Ejercicio 1

## Seudocódigo Recorrido-Niveles

```
q: cola de vértices;  
encolar raíz R en q;  
mientras (cola no se vacíe) {  
    desencolar v de q;  
    imprimir (dato de v);  
    para cada hijo w de v  
        encolar w en q; }  
}
```

**v**  
↑  
~~A~~  
A



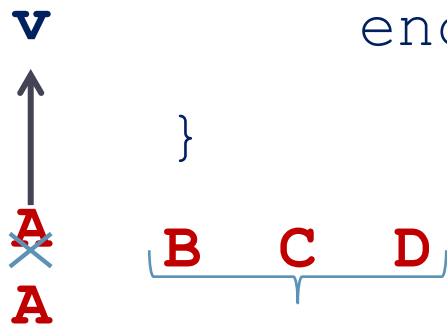
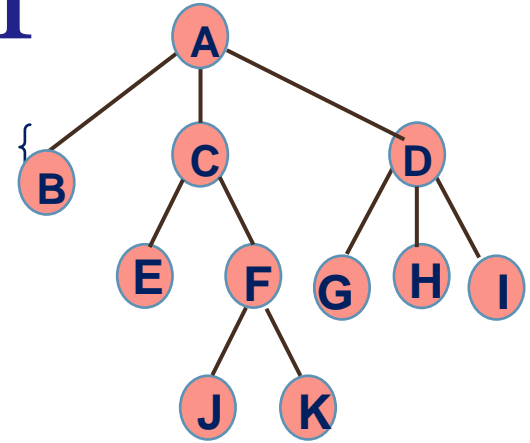
← Cola

← Salida

# Resolución del Ejercicio 1

## Seudocódigo Recorrido-Niveles

```
q: cola de vértices;  
encolar raíz R en q;  
mientras (cola no se vacíe) {  
    desencolar v de q;  
    imprimir (dato de v);  
    para cada hijo w de v  
        encolar w en q; }  
}
```



← Cola  
← Salida

# Resolución del Ejercicio 1

## Seudocódigo Recorrido-Niveles

q: cola de vértices;

encolar raíz **R** en q;

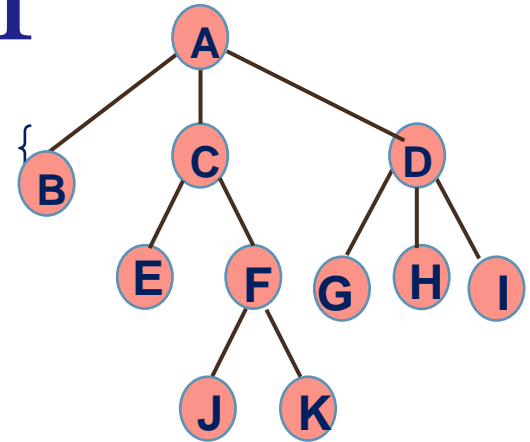
mientras (cola no se vacíe) {

    desencolar **v** de q;

    imprimir (**dato** de v);

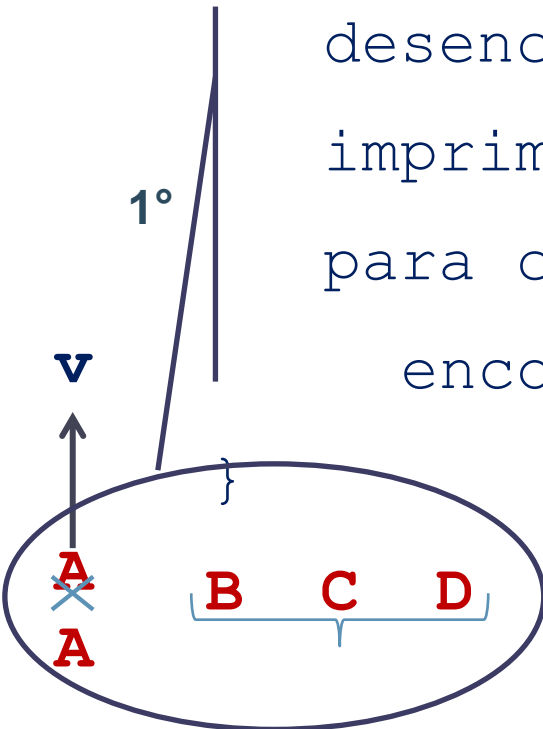
    para cada hijo **w** de **v**

        encolar **w** en q; }



← Cola

← Salida

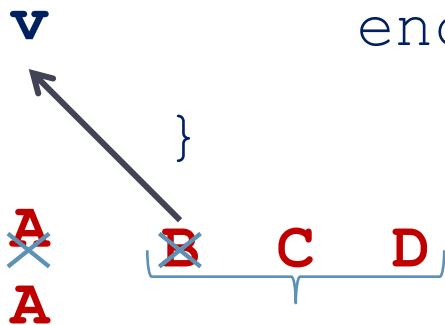
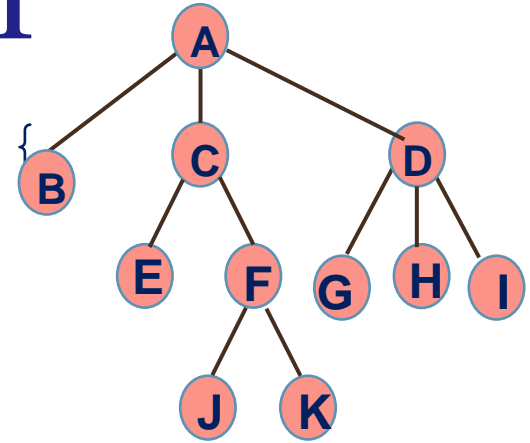




# Resolución del Ejercicio 1

## Seudocódigo Recorrido-Niveles

```
q: cola de vértices;  
encolar raíz R en q;  
mientras (cola no se vacíe) {  
    desencolar v de q;  
    imprimir (dato de v);  
    para cada hijo w de v  
        encolar w en q; }  
}
```



← Cola  
← Salida

# Resolución del Ejercicio 1

## Seudocódigo Recorrido-Niveles

q: cola de vértices;

encolar raíz **R** en q;

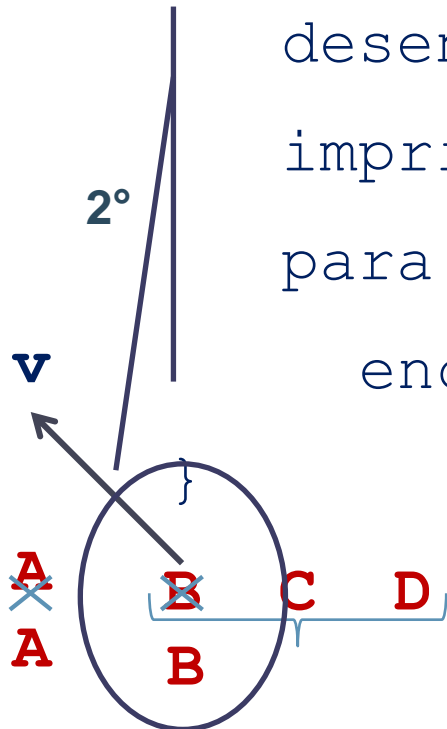
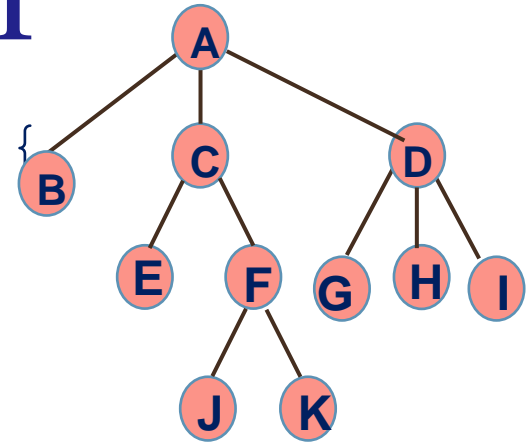
mientras (cola no se vacíe) {

    desencolar **v** de q;

    imprimir (**dato** de v);

    para cada hijo **w** de **v**

        encolar **w** en q; }



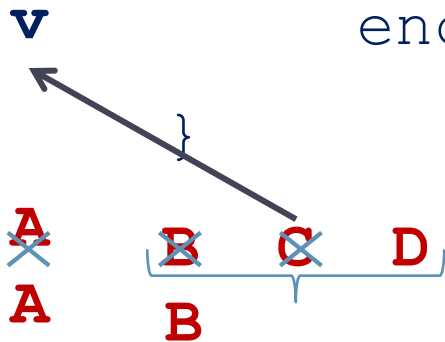
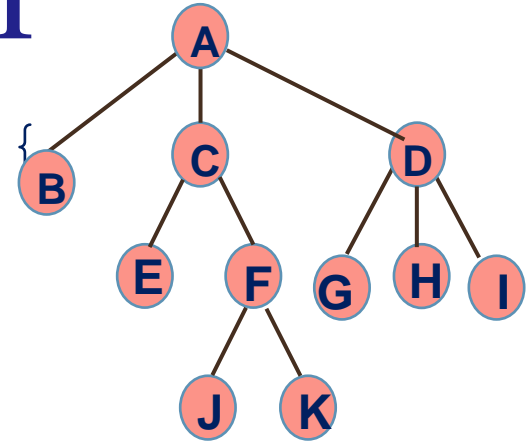
← Cola

← Salida

# Resolución del Ejercicio 1

## Seudocódigo Recorrido-Niveles

```
q: cola de vértices;  
encolar raíz R en q;  
mientras (cola no se vacíe) {  
    desencolar v de q;  
    imprimir (dato de v);  
    para cada hijo w de v  
        encolar w en q; }  
}
```

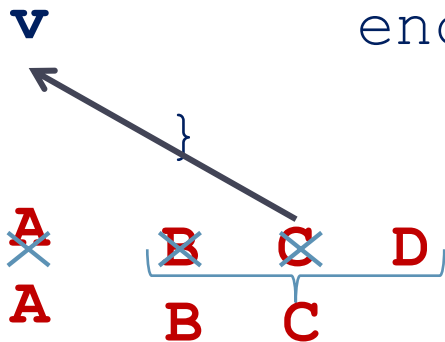
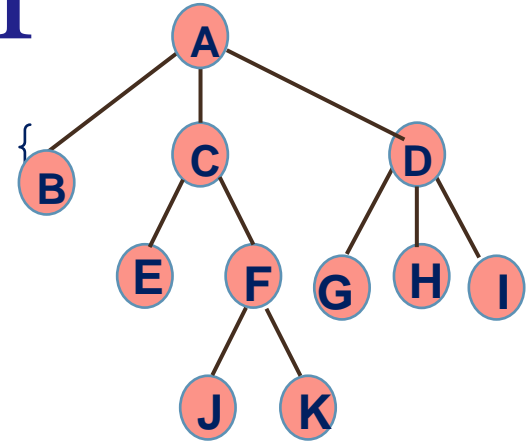


← Cola  
← Salida

# Resolución del Ejercicio 1

## Seudocódigo Recorrido-Niveles

```
q: cola de vértices;  
encolar raíz R en q;  
mientras (cola no se vacíe) {  
    desencolar v de q;  
    imprimir (dato de v);  
    para cada hijo w de v  
        encolar w en q; }
```



← Cola  
← Salida

# Resolución del Ejercicio 1

## Seudocódigo Recorrido-Niveles

q: cola de vértices;

encolar raíz **R** en q;

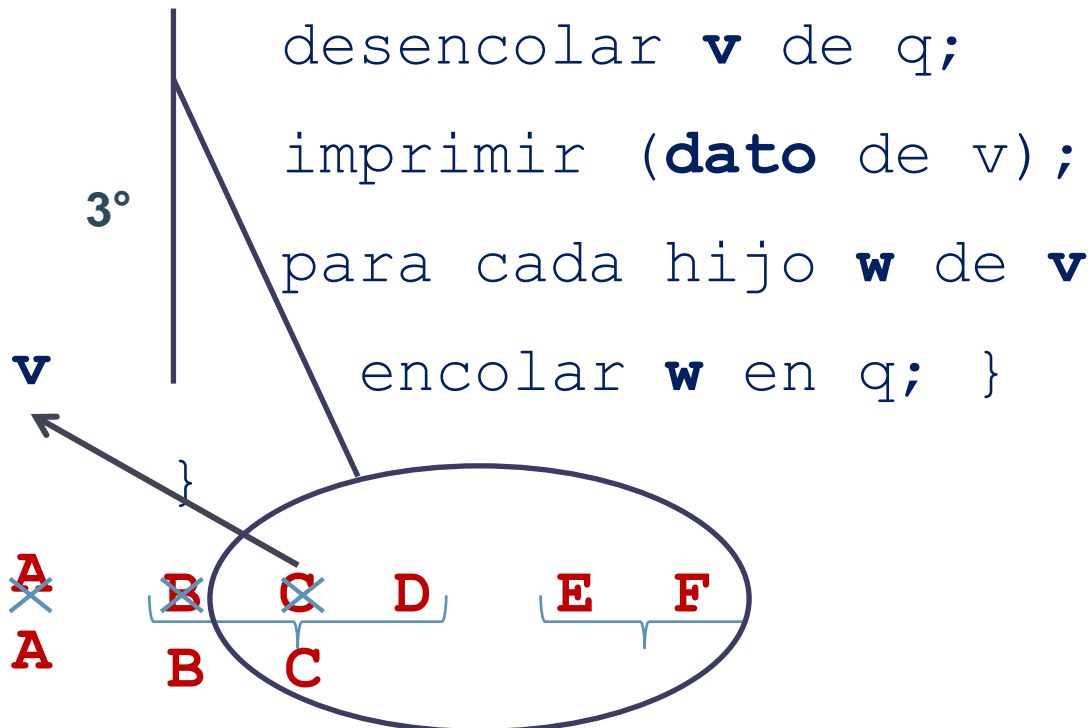
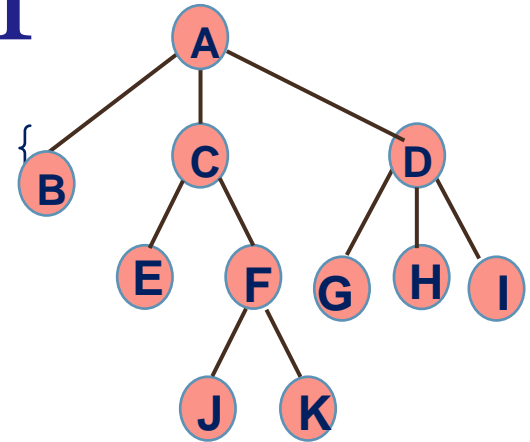
mientras (cola no se vacíe) {

    desencolar **v** de q;

    imprimir (**dato** de v);

    para cada hijo **w** de **v**

        encolar **w** en q; }



← Cola

← Salida

# Resolución del Ejercicio 1

## Seudocódigo Recorrido-Niveles

q: cola de vértices;

encolar raíz **R** en q;

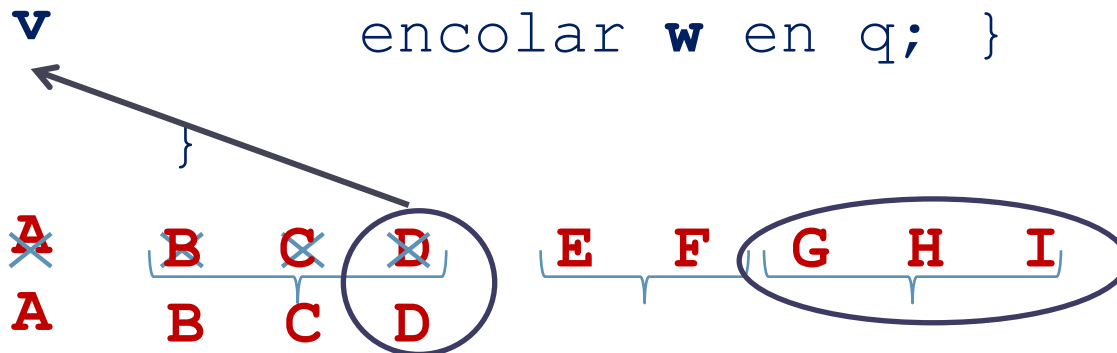
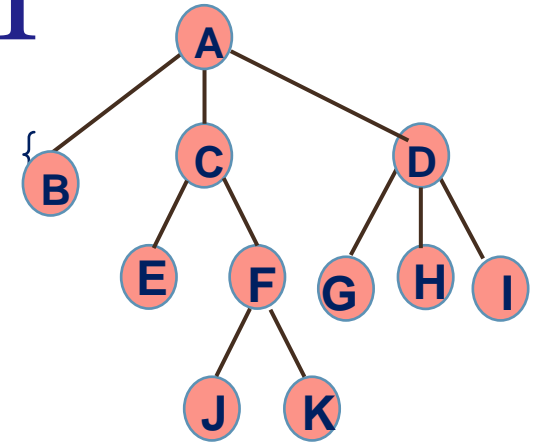
mientras (cola no se vacíe) {

    desencolar **v** de q;

    imprimir (**dato** de v);

    para cada hijo **w** de **v**

        encolar **w** en q; }



← Cola

← Salida

# Resolución del Ejercicio 1

## Seudocódigo Recorrido-Niveles

q: cola de vértices;

encolar raíz **R** en q;

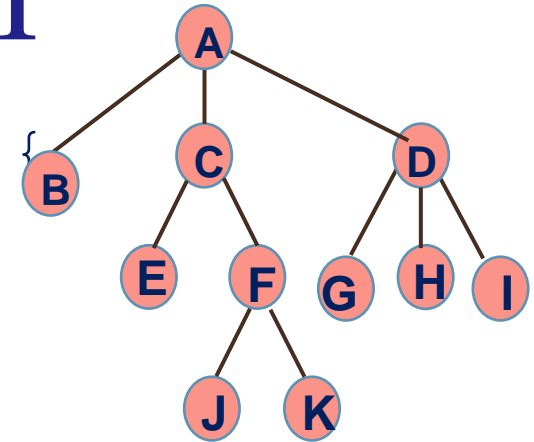
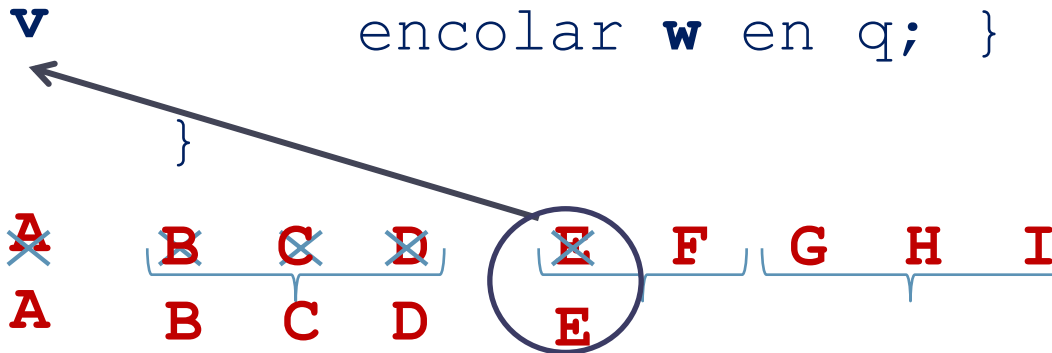
mientras (cola no se vacíe) {

    desencolar **v** de q;

    imprimir (**dato** de v);

    para cada hijo **w** de **v**

        encolar **w** en q; }



← Cola

← Salida

# Resolución del Ejercicio 1

## Seudocódigo Recorrido-Niveles

q: cola de vértices;

encolar raíz **R** en q;

mientras (cola no se vacíe) {

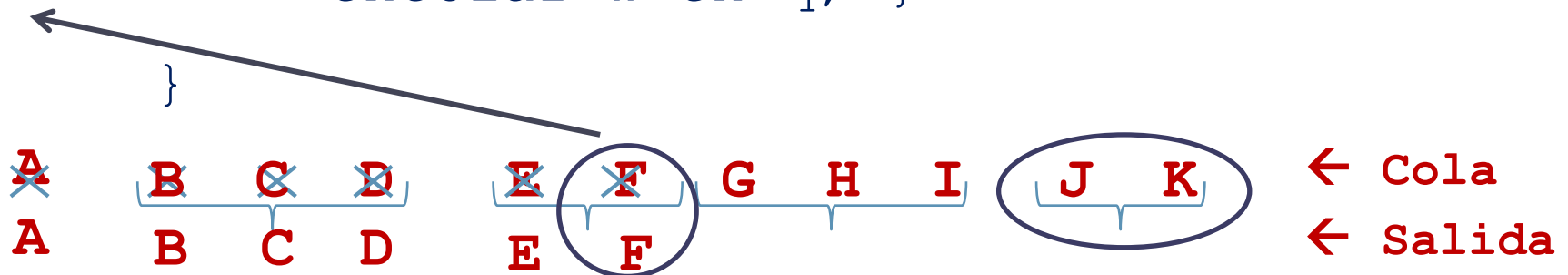
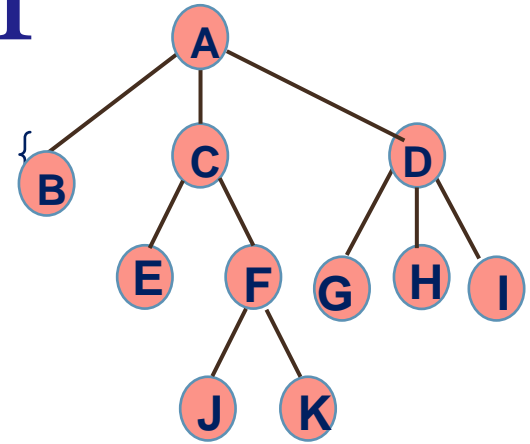
    desencolar **v** de q;

    imprimir (**dato** de v);

    para cada hijo **w** de **v**

        encolar **w** en q; }

**v**





# Resolución del Ejercicio 1

## Seudocódigo Recorrido-Niveles

q: cola de vértices;

encolar raíz **R** en q;

mientras (cola no se vacíe) {

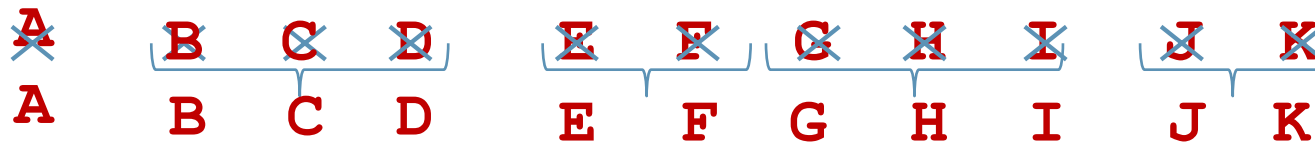
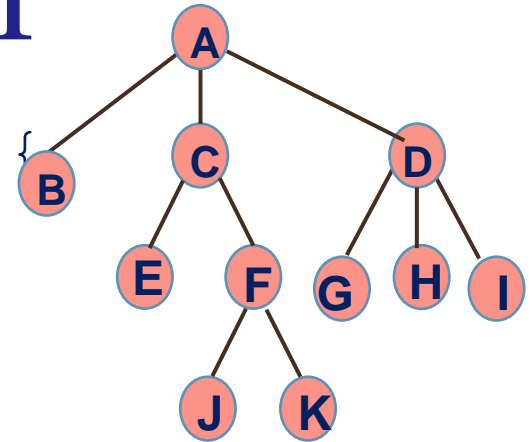
    desencolar **v** de q;

    imprimir (**dato** de v);

    para cada hijo **w** de **v**

        encolar **w** en q; }

}

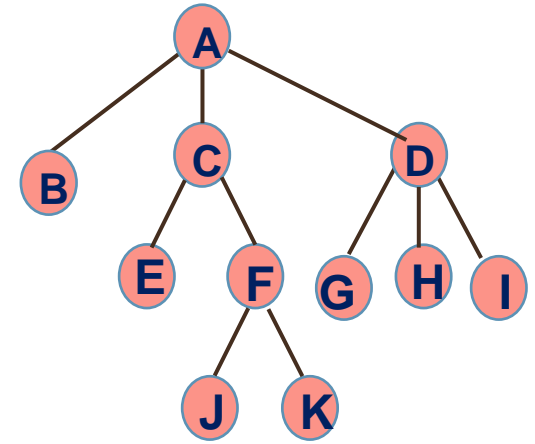


← Cola

← Salida

# Recorrido por niveles

```
Seudocódigo Recorrido_Niveles {  
    q: cola de vértices;  
    encolar raíz R en q;  
    mientras (cola no se vacíe) {  
        desencolar v de q;  
        imprimir (dato de v);  
        para cada hijo w de v  
            encolar w en q;  
    }  
}
```



**A B C D E F G H I J K**

# Resolución del Ejercicio 1

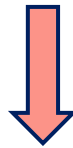
¿Cuántos **niveles** tiene el árbol?

- En el recorrido por niveles no se distingue a qué nivel pertenece cada nodo
- Necesito llevar un control de los niveles

# Resolución del Ejercicio 1

¿Cuántos **niveles** tiene el árbol?

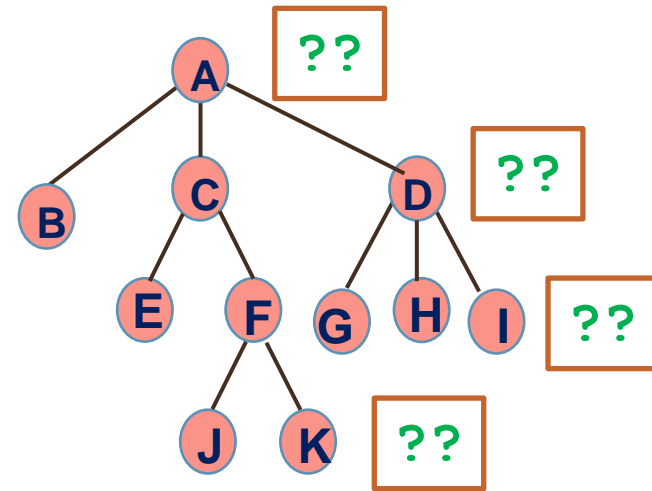
- En el recorrido por niveles no se distingue a qué nivel pertenece cada nodo
- Necesito llevar un control de los niveles



Inserto una marca al finalizar cada nivel

# Resolución del Ejercicio 1

```
Seudocódigo Ejerc1-Niveles {  
    q: cola de vértices;  
    encolar raíz R en q;  
    mientras (cola no se vacíe) {  
        desencolar v de q;  
        imprimir (dato de v);  
        para cada hijo w de v  
            encolar w en q; }  
}
```



# Resolución del Ejercicio 1

## Seudocódigo Ejerc1-Niveles {

q: cola de vértices;

encolar raíz **R** en q;

mientras (cola no se vacíe) {

    desencolar **v** de q;

    imprimir (dato de v);

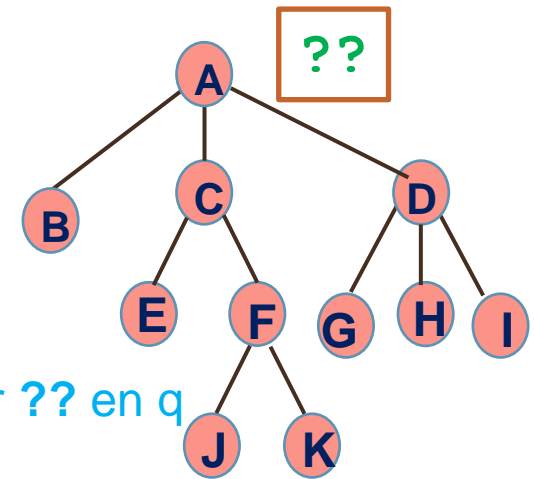
    para cada hijo **w** de **v**

        encolar **w** en q; }

}

??

**A**



encolar ?? en q

← Cola

# Resolución del Ejercicio 1

## Seudocódigo Ejerc1-Niveles {

q: cola de vértices;

encolar raíz **R** en q;

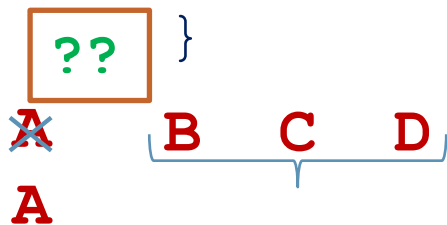
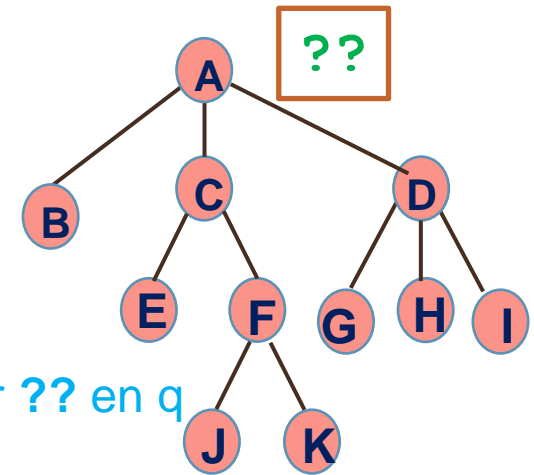
mientras (cola no se vacíe) {

    desencolar **v** de q;

    imprimir (dato de v);

    para cada hijo **w** de **v**

        encolar **w** en q; }



← Cola

← Salida

# Resolución del Ejercicio 1

## Seudocódigo Ejerc1-Niveles {

q: cola de vértices;

encolar raíz **R** en q;

mientras (cola no se vacíe) {

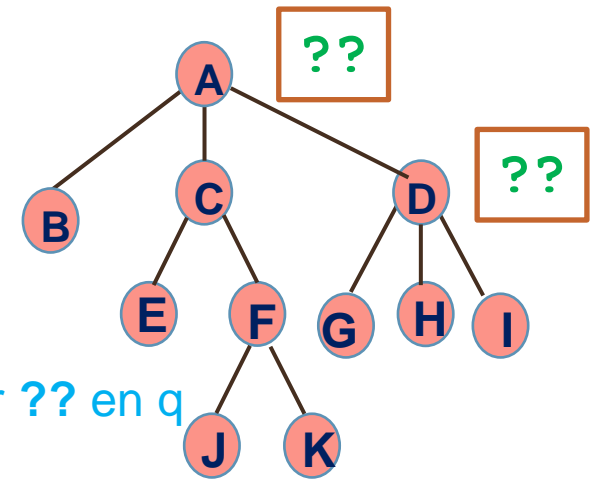
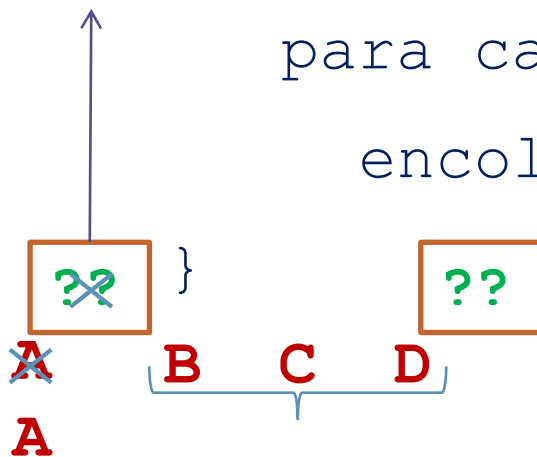
desencolar **v** de q;

imprimir (dato de v);

para cada hijo **w** de **v**

encolar **w** en q; }

Fin de  
nivel



← Cola

← Salida



# Resolución del Ejercicio 1

## Seudocódigo Ejerc1-Niveles {

q: cola de vértices;

encolar raíz **R** en q;

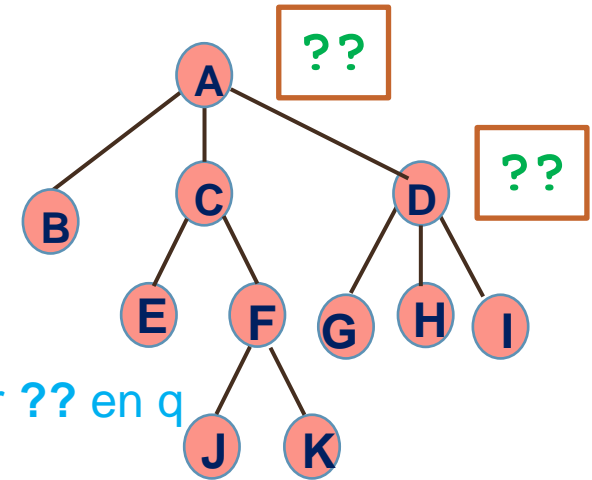
mientras (cola no se vacíe) {

desencolar **v** de q;

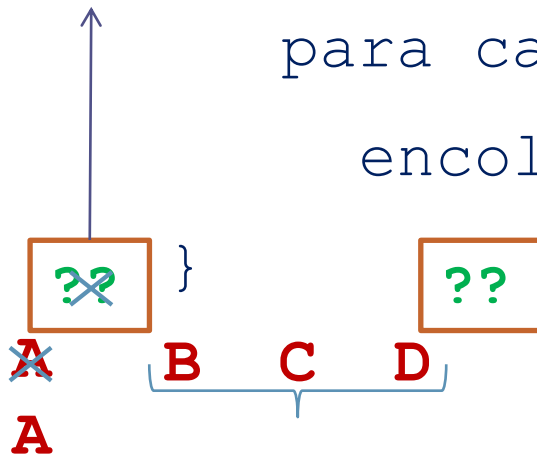
imprimir (dato de v);

para cada hijo **w** de **v**

encolar **w** en q; }



Fin de  
nivel



encolar ?? en q

si (dato de v es ??) encolar ?? en q

Ojo ! Podría imprimir ??

← Cola

← Salida

# Resolución del Ejercicio 1

## Seudocódigo Ejerc1-Niveles {

q: cola de vértices;

encolar raíz **R** en q;

mientras (cola no se vacíe) {

    desencolar **v** de q;

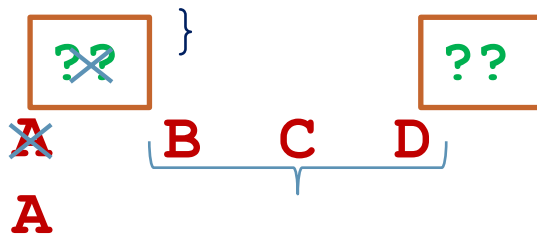
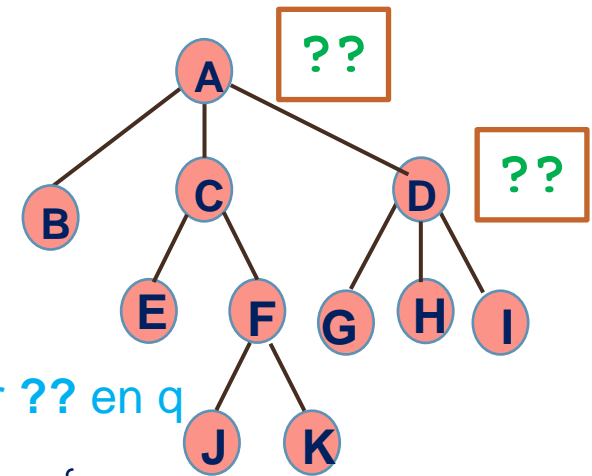
    si (dato de v es ??) encolar ?? en q

    sino

        imprimir (dato de v);

    para cada hijo **w** de **v**

        encolar **w** en q; }



← Cola

← Salida

# Resolución del Ejercicio 1

## Seudocódigo Ejerc1-Niveles {

q: cola de vértices;

encolar raíz **R** en q;

mientras (cola no se vacíe) {

    desencolar **v** de q;

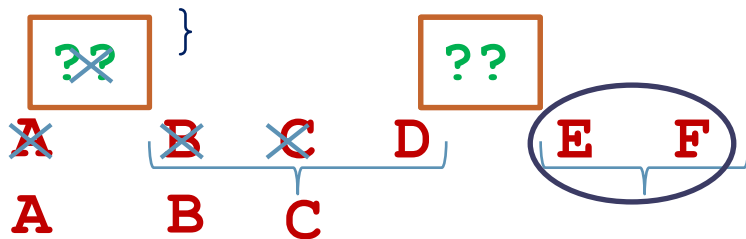
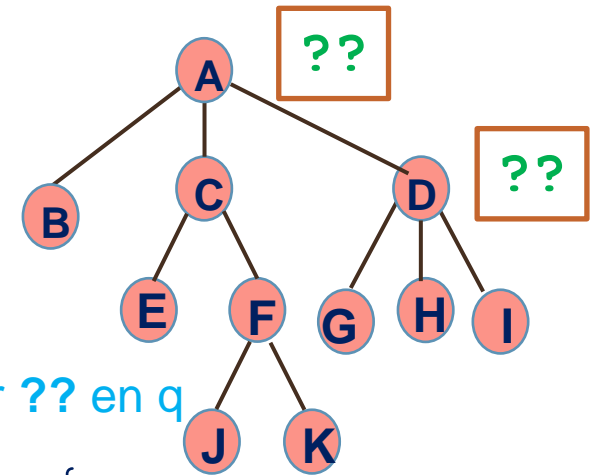
    si (dato de v es ??) encolar ?? en q

    sino

        imprimir (dato de v);

    para cada hijo **w** de **v**

        encolar **w** en q; }



← Cola

← Salida

# Resolución del Ejercicio 1

## Seudocódigo Ejerc1-Niveles {

q: cola de vértices;

encolar raíz **R** en q;

mientras (cola no se vacíe) {

    desencolar **v** de q;

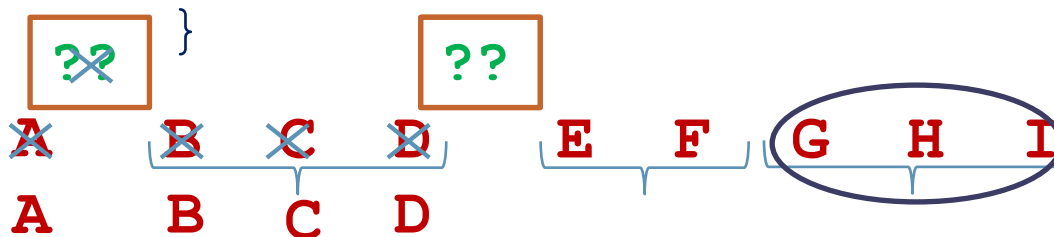
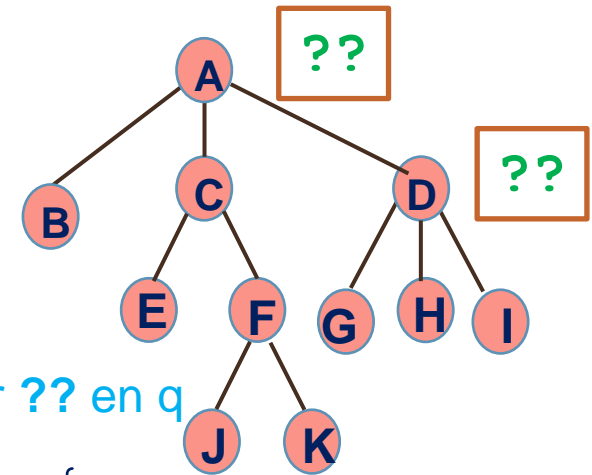
    si (dato de v es ??) encolar ?? en q

    sino

        imprimir (dato de v);

    para cada hijo **w** de **v**

        encolar **w** en q; }



← Cola

← Salida

# Resolución del Ejercicio 1

## Seudocódigo Ejerc1-Niveles {

q: cola de vértices;

encolar raíz **R** en q;

mientras (cola no se vacíe) {

    desencolar **v** de q;

    si (dato de v es ??) encolar ?? en q

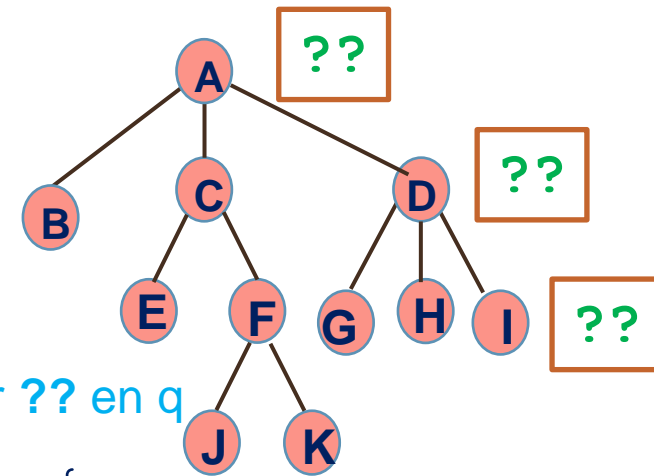
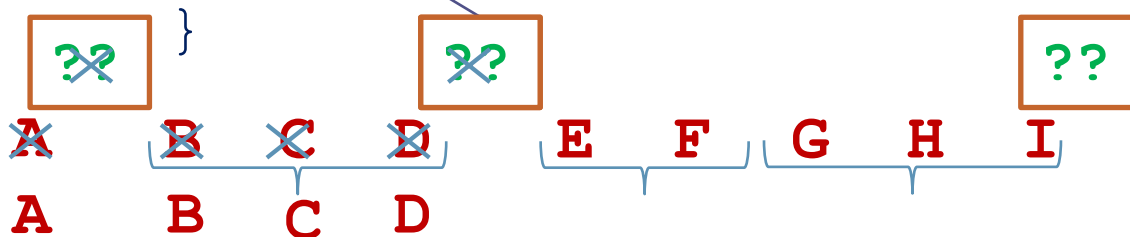
    sino

        imprimir (dato de v);

        para cada hijo **w** de **v**

            encolar **w** en q; }

Fin de  
nivel



← Cola

← Salida

# Resolución del Ejercicio 1

## Seudocódigo Ejerc1-Niveles {

q: cola de vértices;

encolar raíz **R** en q;

mientras (cola no se vacíe) {

    desencolar **v** de q;

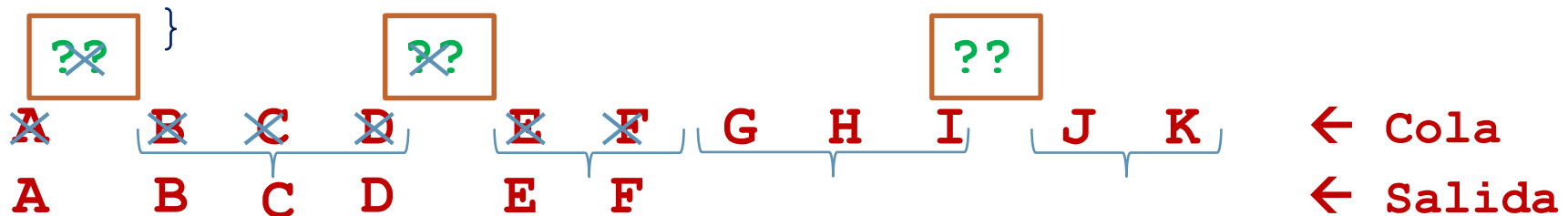
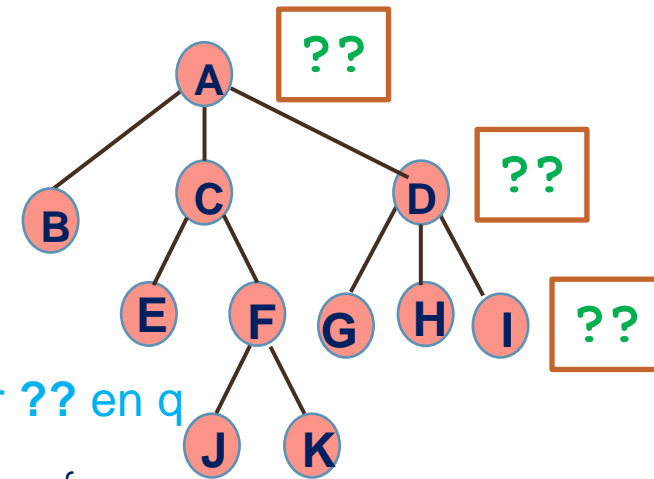
    si (dato de v es ??) encolar ?? en q

    sino

        imprimir (dato de v);

    para cada hijo **w** de **v**

        encolar **w** en q; }



# Resolución del Ejercicio 1

## Seudocódigo Ejerc1-Niveles {

q: cola de vértices;

encolar raíz **R** en q;

mientras (cola no se vacíe) {

    desencolar **v** de q;

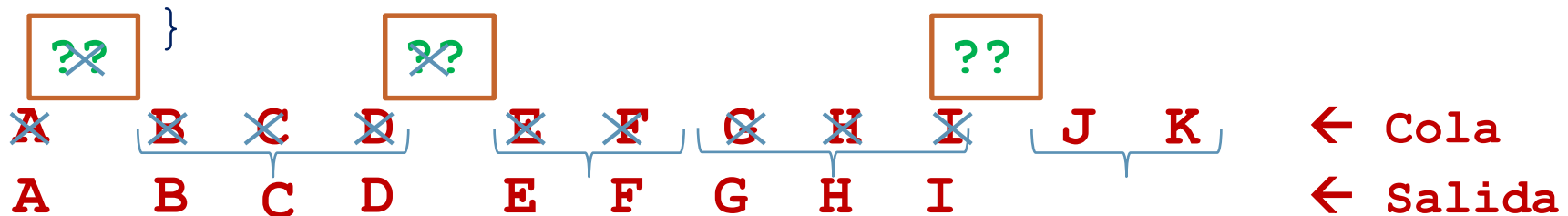
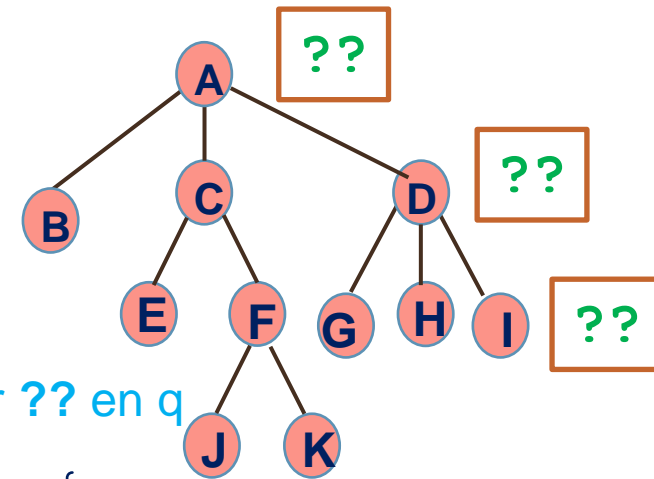
    si (dato de v es ??) encolar ?? en q

    sino

        imprimir (dato de v);

    para cada hijo **w** de **v**

        encolar **w** en q; }



# Resolución del Ejercicio 1

## Seudocódigo Ejerc1-Niveles {

q: cola de vértices;

encolar raíz **R** en q;

mientras (cola no se vacíe) {

    desencolar **v** de q;

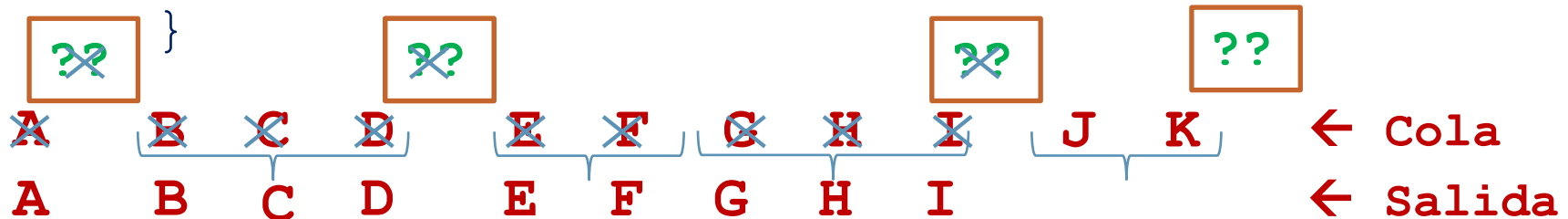
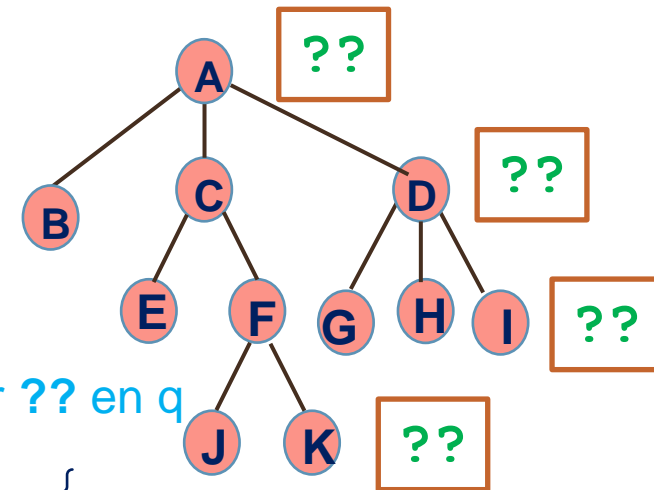
    si (dato de v es ??) encolar ?? en q

    sino

        imprimir (dato de v);

    para cada hijo **w** de **v**

        encolar **w** en q; }





# Resolución del Ejercicio 1

## Seudocódigo Ejerc1-Niveles {

q: cola de vértices;

encolar raíz **R** en q;

mientras (cola no se vacíe) {

    desencolar **v** de q;

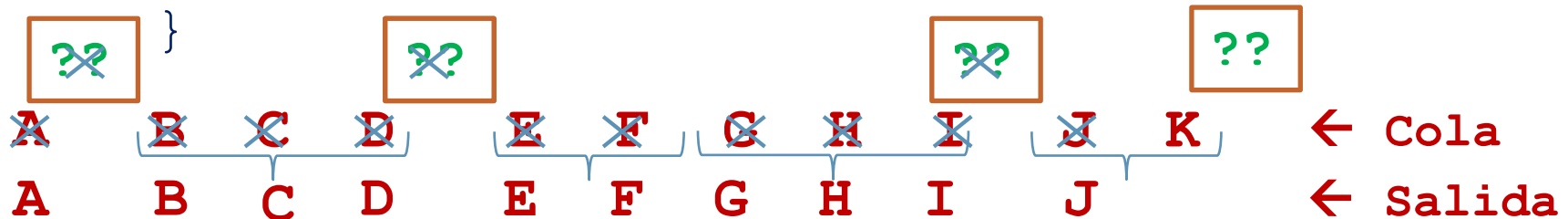
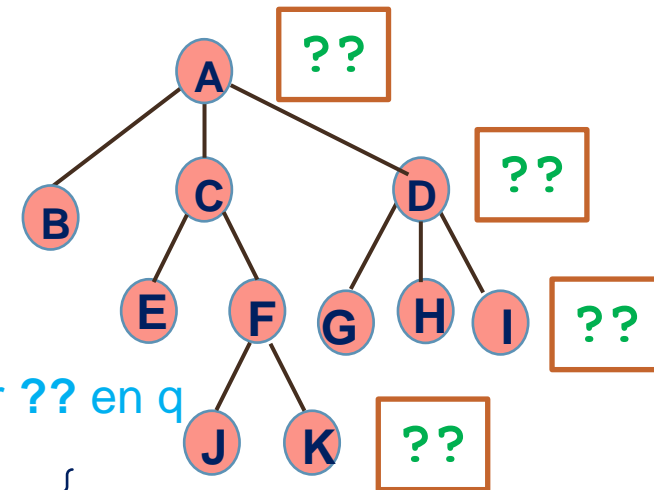
    si (dato de v es ??) encolar ?? en q

    sino

        imprimir (dato de v);

    para cada hijo **w** de **v**

        encolar **w** en q; }



# Resolución del Ejercicio 1

## Seudocódigo Ejerc1-Niveles {

q: cola de vértices;

encolar raíz **R** en q;

mientras (cola no se vacíe) {

    desencolar **v** de q;

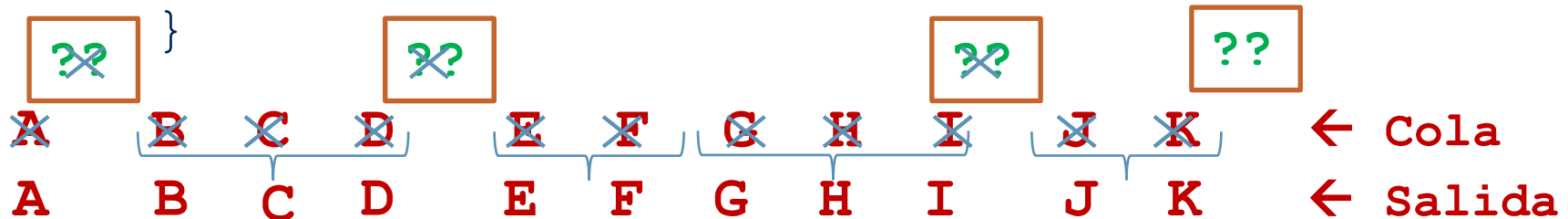
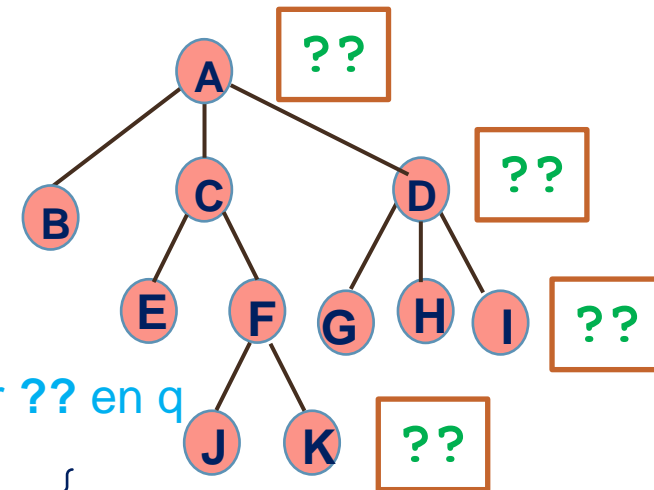
    si (dato de v es ??) encolar ?? en q

    sino

        imprimir (dato de v);

    para cada hijo **w** de **v**

        encolar **w** en q; }



# Resolución del Ejercicio 1

## Seudocódigo Ejerc1-Niveles {

q: cola de vértices;

encolar raíz **R** en q;

mientras (cola no se vacíe) {

    desencolar **v** de q;

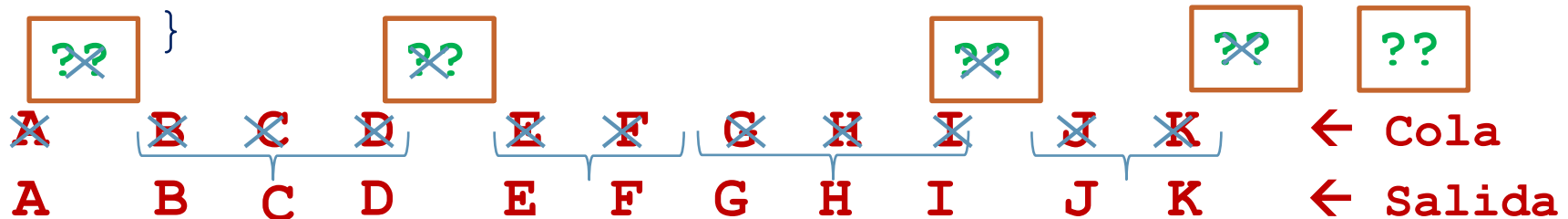
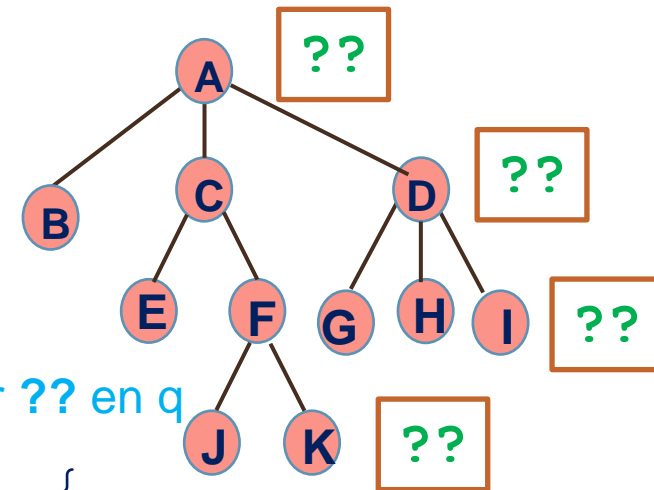
    si (dato de v es ??) encolar ?? en q

    sino

        imprimir (dato de v);

    para cada hijo **w** de **v**

        encolar **w** en q; }



# Resolución del Ejercicio 1

## Seudocódigo Ejerc1-Niveles {

q: cola de vértices;

encolar raíz **R** en q;

mientras (cola no se vacíe) {

    desencolar **v** de q;

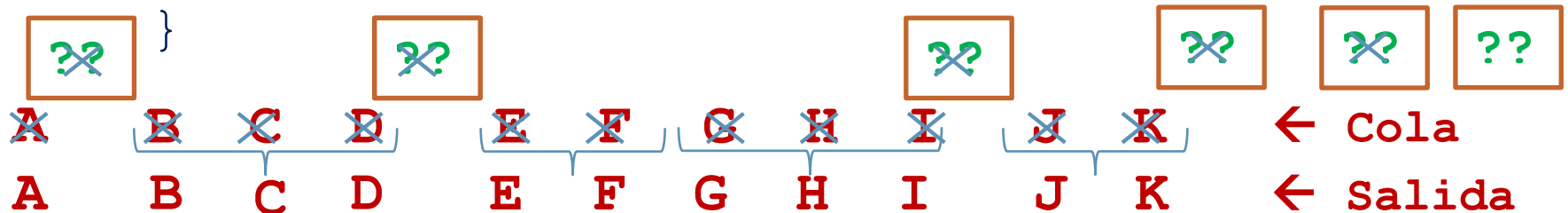
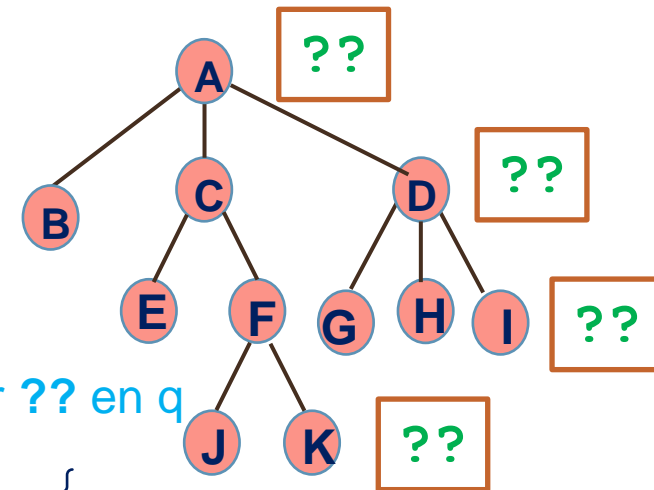
    si (dato de v es ??) encolar ?? en q

    sino

        imprimir (dato de v);

    para cada hijo **w** de **v**

        encolar **w** en q; }



# Resolución del Ejercicio 1

## Seudocódigo Ejerc1-Niveles {

q: cola de vértices;

encolar raíz **R** en q;

mientras (cola no se vacíe) {

    desencolar **v** de q;

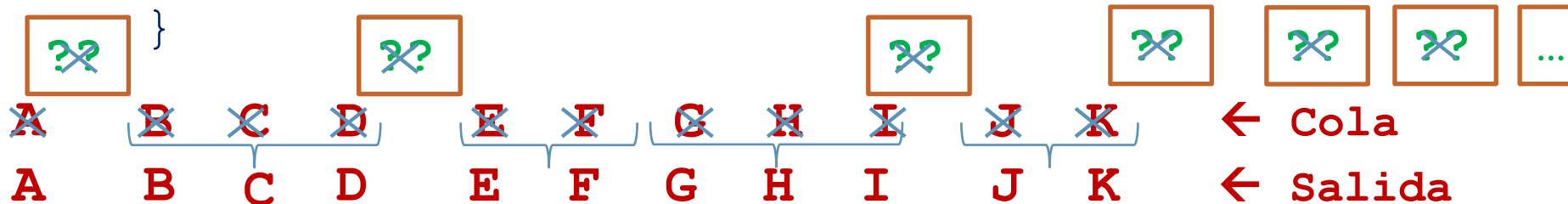
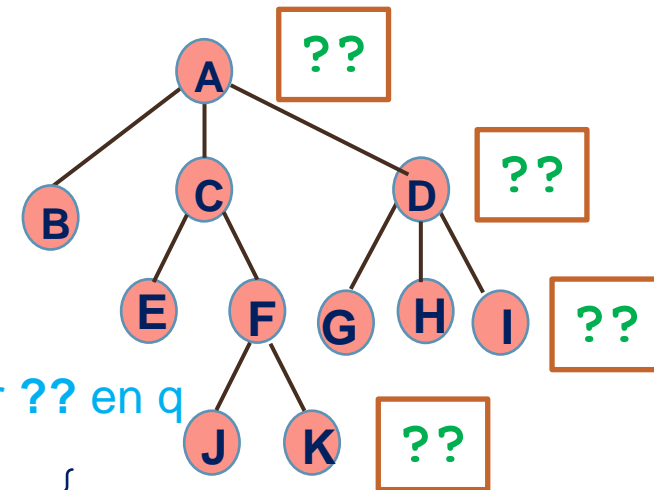
    si (dato de v es ??) encolar ?? en q

    sino

        imprimir (dato de v);

    para cada hijo **w** de **v**

        encolar **w** en q; }



# Resolución del Ejercicio 1

## Seudocódigo Ejerc1-Niveles {

q: cola de vértices;

encolar raíz **R** en q; **encolar ?? en q;**

mientras (cola no se vacíe) {

desencolar **v** de q;

si (dato de **v** es ??

encolar ?? en q

sino

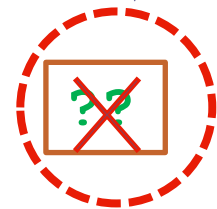
si (dato de **v** no es ??)

imprimir (dato de **v**);

para cada hijo **w** de **v**

encolar **w** en q; }

}



**A**

**B**

**C**

**D**

**E**

**F**

**G**

**H**

**I**

**J**

**K**

para evitar  
ésto.

# Resolución del Ejercicio 1

## Seudocódigo Ejerc1-Niveles {

q: cola de vértices;

encolar raíz **R** en q; encolar ?? en q;

mientras (cola no se vacíe) {

    desencolar **v** de q;

    si (dato de **v** es ?? & q no está vacía)

        encolar ?? en q;

    sino

        si (dato de **v** no es ??) {

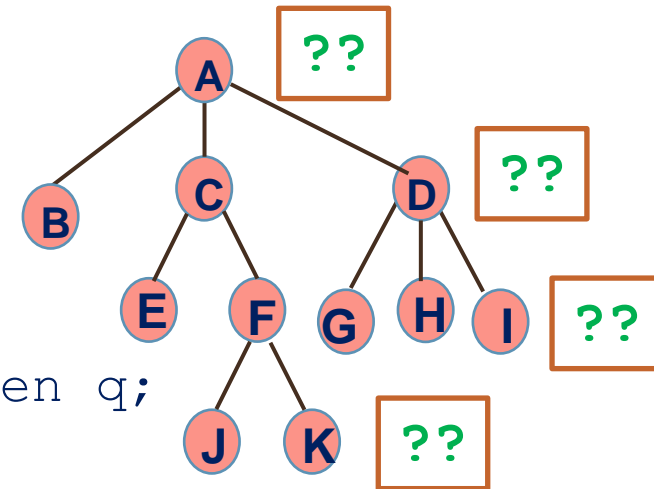
            imprimir (dato de **v**);

            para cada hijo **w** de **v**

                encolar **w** en q; }

}

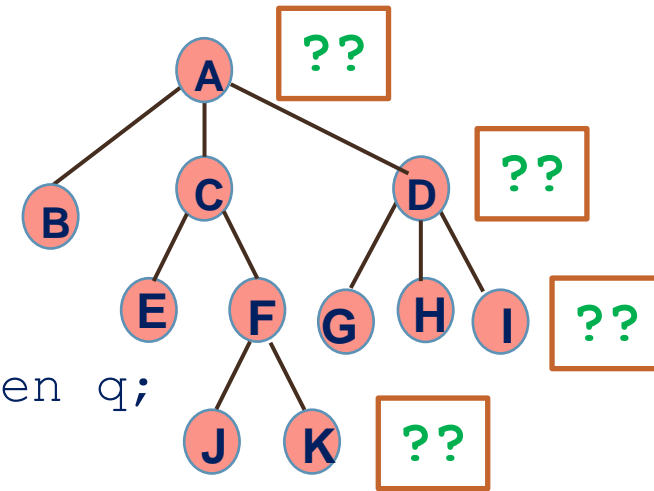
}



# Resolución del Ejercicio 1

## Seudocódigo Ejerc1-Niveles {

```
q: cola de vértices;  
encolar raíz R en q; encolar ?? en q;  
mientras (cola no se vacíe) {  
    desencolar v de q;  
    si (dato de v no es ??) {  
        imprimir (dato de v);  
        para cada hijo w de v  
            encolar w en q; }  
    sino  
        si (q no está vacía)  
            encolar ?? en q;  
}  
}
```





# Resolución del Ejercicio 1

## Seudocódigo Ejerc1-Niveles {

q: cola de vértices;

encolar raíz **R** en q; encolar ?? en q;

← cantNiveles = 0;

mientras (cola no se vacíe) {

desencolar **v** de q;

si (dato de **v** no es ??) {

imprimir (dato de **v**);

para cada hijo **w** de **v**

encolar **w** en q; }

sino

si (q no está vacía)

encolar ?? en q;

← cantNiveles ++;

}

← return cantNiveles++;

}

# Resolución del Ejercicio 2

## Seudocódigo Ejerc2-Niveles {

q: cola de vértices;

encolar raíz **R** en q; encolar ?? en q;

← nroNivel = 0;

mientras (cola no se vacíe) {

desencolar **v** de q;

si (dato de **v** no es ??) {

imprimir (dato de **v**);

para cada hijo **w** de **v**

encolar **w** en q; }

sino

si (q no está vacía)

encolar ?? en q;

← nroNivel ++;

}

}

# Resolución del Ejercicio 2

## Seudocódigo Ejerc2-Niveles {

```
q: cola de vértices;           ← cantNodos: array de enteros;
encolar raíz R en q; encolar ?? en q;           ← nroNivel = 0;
mientras (cola no se vacíe) {
    desencolar v de q;
    si (dato de v no es ??) {
        imprimir (dato de v);
        para cada hijo w de v           ← cantNodos[nroNivel]++;
            encolar w en q; }
    sino
        si (q no está vacía)
            encolar ?? en q;
        }
    }
} ← return cantNodos;
```

# Resolución del Ejercicio 3

**Seudocódigo Ejerc3-Niveles** (int k) {

q: cola de vértices;

← nroNivel = 0;

encolar raíz **R** en q; encolar ?? en q;

← cantNodos = 0;

mientras (cola no se vacíe) {

desencolar **v** de q;

si (dato de **v** no es ??) {

imprimir (dato de **v**);

← si (nroNivel == k)

{ /\*Contar la cantidad de  
nodos en el nivel k \*/  
cantNodos++;

para cada hijo **w** de **v**

encolar **w** en q; }

sino

si (q no está vacía) {

encolar ?? en q;

← nroNivel ++ ;

}

← return cantNodos;

}

# Resolución del Ejercicio 3

```
Seudocódigo Ejerc3-Niveles (int k) {  
    q: cola de vértices; nroNivel=0; cantNodos=0;  
    encolar raíz R en q; encolar ?? en q;  
    mientras (cola no se vacíe) {  
        desencolar v de q;  
        si (dato de v no es ??)  
            si (nroNivel==k)  
                mientras (dato de v no es ??)  
                    {cantNodos++;  
                     desencolar v de q;}  
            sino para cada hijo w de v  
                encolar w en q;  
    sino  
        si (q no está vacía){  
            encolar ?? en q;  
            nroNivel++; }  
    }  
    return cantNodos;  
}
```

# Ejercicio

## Abeto navideño

[Problem - B - Codeforces](#)

Considere un árbol general. Recordemos que el vértice  $u$  se llama hijo del vértice  $v$  y el vértice  $v$  se llama padre del vértice  $u$  si existe una arista dirigida de  $v$  a  $u$ . El árbol tiene un vértice distinguido llamado **raíz**, que es el único vértice que no tiene padre. Un vértice se llama **hoja** si no tiene hijos y tiene padre.

Llamaremos **abeto** a un árbol si cada vértice no hoja tiene al menos 3 hijos hojas. Dado un árbol general, compruebe si es un abeto.

### Input

La primera línea contiene un entero  $n$ : el número de vértices en el árbol ( $3 \leq n \leq 1000$ ). Cada una de las siguientes  $n - 1$  líneas contiene un entero  $p_i$  ( $1 \leq i \leq n - 1$ ) — el índice del padre del  $i + 1$ -ésimo vértice ( $1 \leq p_i \leq i$ ).

El vértice 1 es la raíz. Está garantizado que la raíz tiene al menos 2 hijos.

### Output

Imprima "Yes" si el árbol es un abeto y "No" de lo contrario.

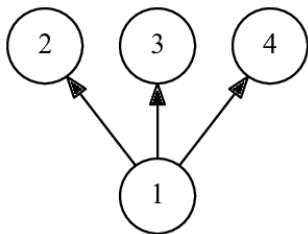
## Ejemplo 1

Input

4  
1  
1  
1

Output

Yes



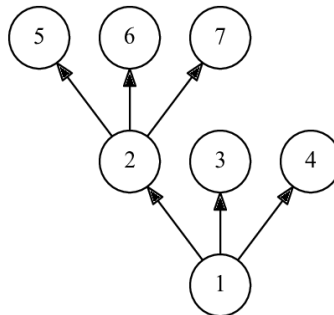
## Ejemplo 2

Input

7  
1  
1  
1  
2  
2  
2

Output

No



## Ejemplo 3

Input

8  
1  
1  
1  
1  
1  
3  
3  
3

Output

Yes

