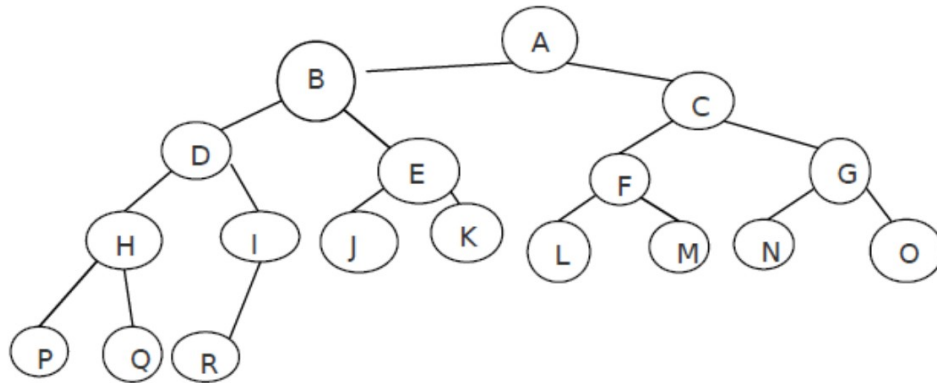
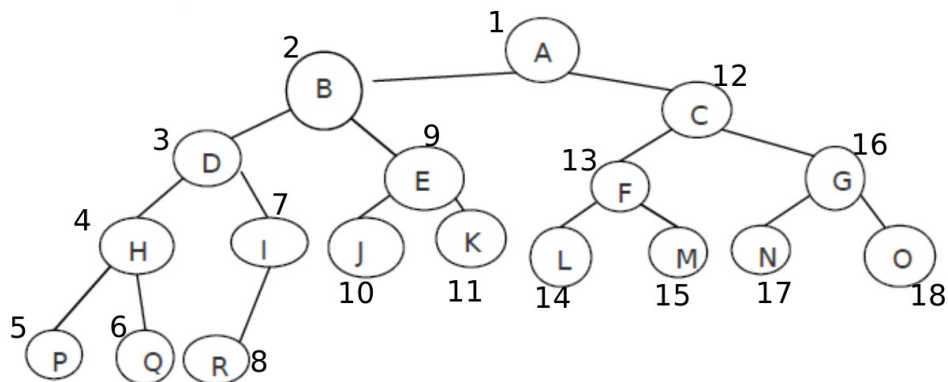


**PROGRAMACIÓN 2 - Unidad 5: Árboles.**  
**Ejercicios entregables de la semana del 30/03 al 3/04**

1. Considérese el árbol siguiente

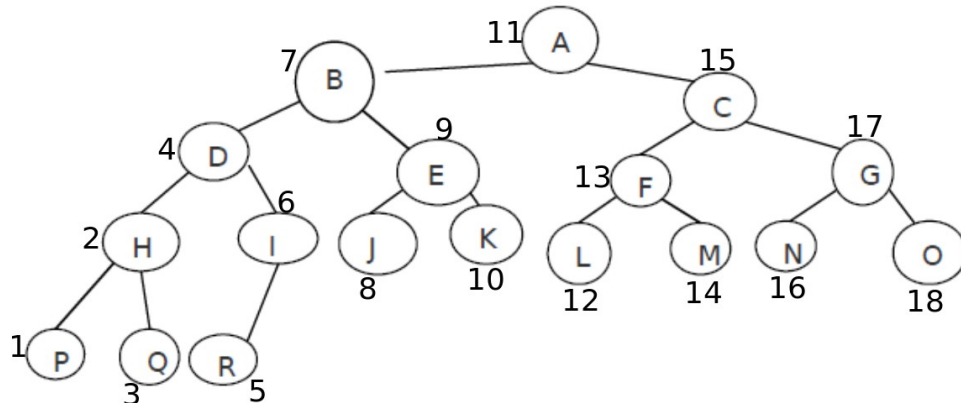


- Su profundidad es 4.
- Es un árbol binario casi-completo; ya que no hay ningún nodo con más de dos nodos hijos, y todos los niveles, excepto el cuarto están completos (y su profundidad es 4).
- El padre del nodo **R** es **I**.
- Los antecesores del nodo **J** son **E**, **B** y **A**.
- Recorrido del árbol según el algoritmo de preorden:



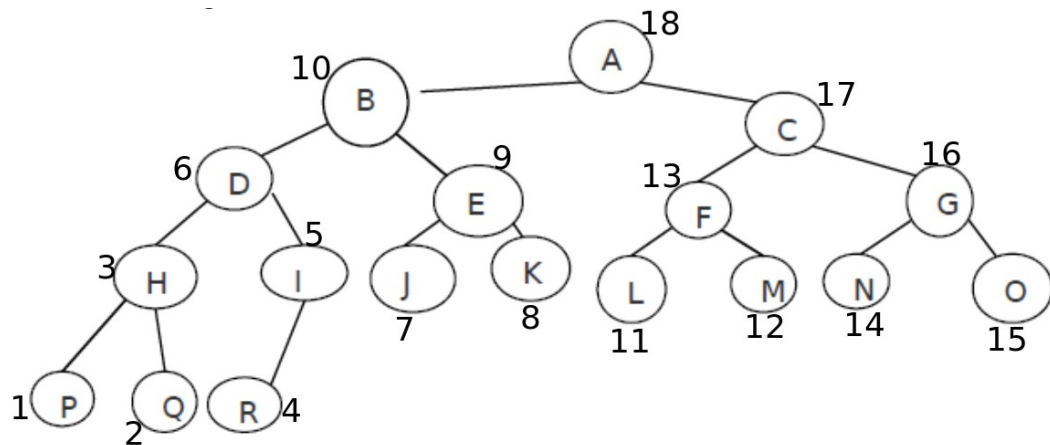
Según orden de visita: **A B D H P Q I R E J K C F L M G N O**

Recorrido del árbol según el algoritmo de orden medio:



Según orden de visita: **P H Q D R I B E J K A L F M C N G O**

Recorrido del árbol según el algoritmo de preorden:



Según orden de visita: **P Q H R I D J K E B L M F N O G C A**

**2. Escriba el pseudocódigo de los algoritmos apropiados para determinar:**

a. El número total de nodos de un árbol binario.

```

Integer num_nodes(Tree t) :
    Integer n = 0

    if t == NULL :
        return 0 //base case

    n = num_nodes(left(t)) //counts the nodes on the left side
    n = n + num_nodes(right(t)) //+ nodes on the right side
    n = n + 1 //+ the node we are in

    return n
  
```

b. El número de hojas de un árbol binario.

```

Integer num_leafs(Tree t) :
    Integer n = 0

    if t == NULL :
        return 0

    if left(t) == NULL AND right(t) == NULL:
        return 1 //this means the node being visited is a leaf node

    n = num_leafs(left(t)) //counts the leafs on the left side
    n = n + num_leafs(right(t)) //+ leafs on the right side

    return n
  
```

c. La suma del contenido de los nodos de un árbol binario de elementos de tipo entero.

```

Integer sum_tree(Tree t) :
    Integer n = 0

    if t == NULL :
        return 0 //base case

    n = sum_tree(left(t))
    n = n + sum_tree(right(t))
    n = n + info(root(t))

    return n
  
```

d. La profundidad de un árbol binario.

(Solución según la definición de profundidad que aparece en las diapositivas)

```
Integer tree_depth(Tree t) :
    Integer depthl, depthr

    if t == NULL :
        return -1 //base case, a tree without nodes

    depthl = tree_depth(left(t))
    depthr = tree_depth(right(t))

    //now we take the maximum of the depths

    if depthl > depthr :
        n = depthl
    else
        n = depthr

    //and add 1

    return n + 1
```