



Estructura de Datos

Maria C. Torres

Maria C. Torres

Ing. Electrónica (UNAL)

M.E. Ing. Eléctrica (UPRM)

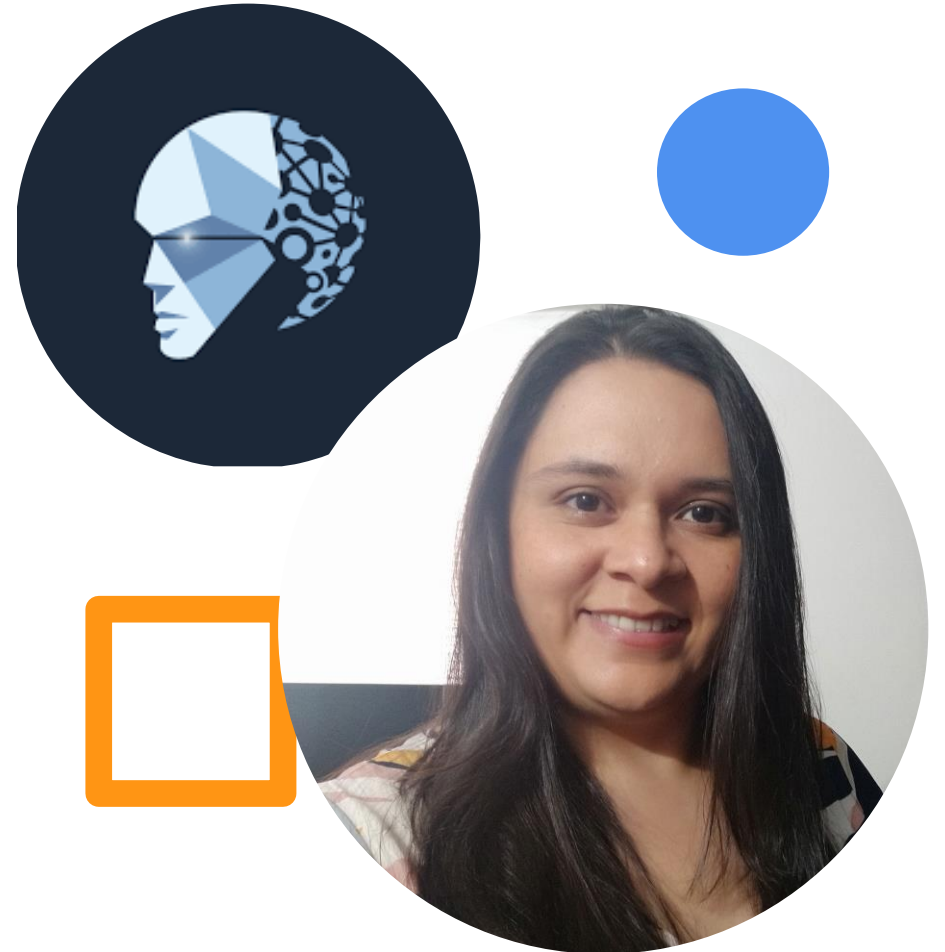
Ph.D. Ciencias e Ingeniería de la Computación y la Información (UPRM)


Profesora asociada

Dpto. Ciencias de la Computación y la Decisión


mctorresm@unal.edu.co

HORARIO DE ATENCIÓN: Martes 10:00 am a
12:00 m – Oficina 313 M8A





Contenido del Curso

- 
- ☐ Introducción: revisión fundamentos y POO
 - ☐ Análisis de complejidad
 - ☐ Arreglos
 - ☐ Listas enlazadas
 - ☐ Pilas y colas
 - ☐ Heap
 - ☐ **Arboles binarios**
 - ☐ Tablas hash
 - ☐ Grafos

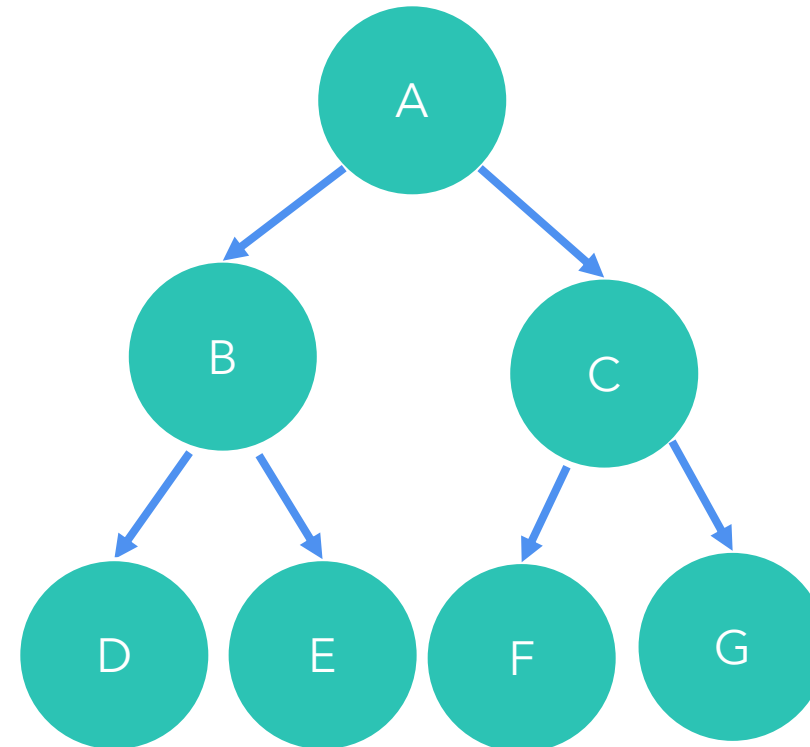


Árbol Binario de Búsqueda

- ☐ Árboles
- ☐ **Árboles binarios**
- ☐ Árboles binarios de búsqueda
- ☐ Árboles balanceados

Arboles – Algoritmos de Recorrido

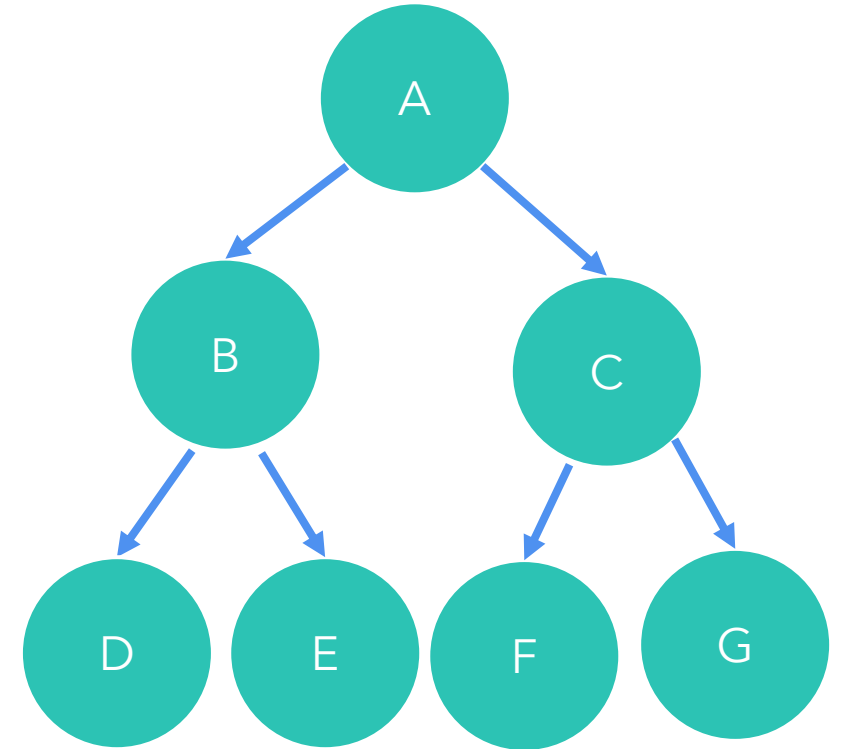
- ❑ Los algoritmos de recorrido de un árbol T son una forma sistemática de visitar o acceder TODOS los nodos del árbol T
- ❑ Existen diferentes formas de acceder o visitar un árbol binario, aquí estudiaremos tres esquemas
 - Pre-orden
 - In-orden
 - Post-orden



Arboles – Algoritmos de Recorrido

Algoritmo pre-orden:

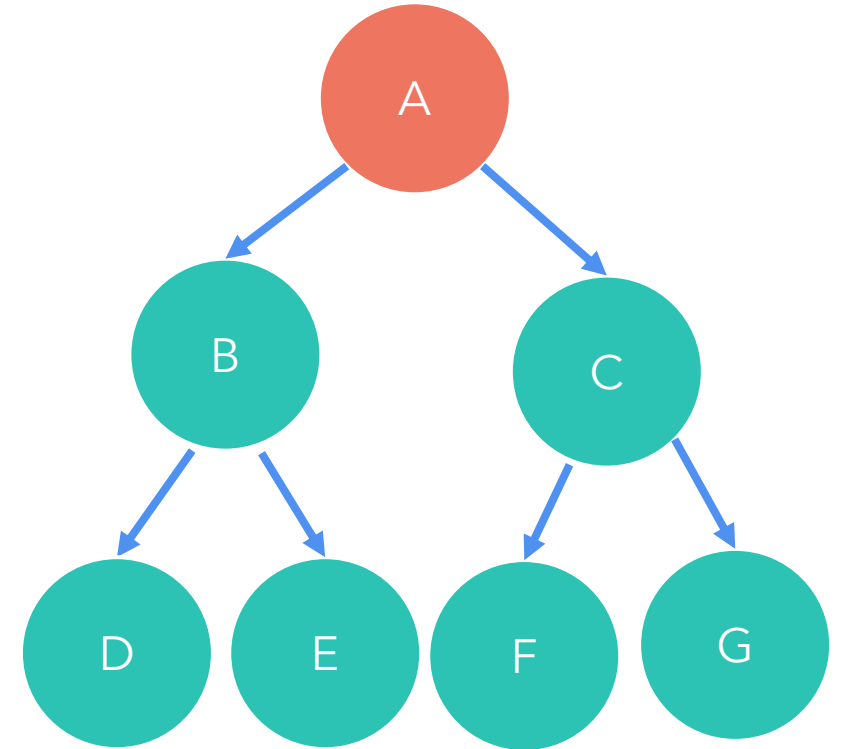
- ❑ En este recorrido primero se visita la raíz
- ❑ Posteriormente, se visita recursivamente el subárbol izquierdo y el subárbol derecho



Arboles – Algoritmos de Recorrido

Algoritmo pre-orden:

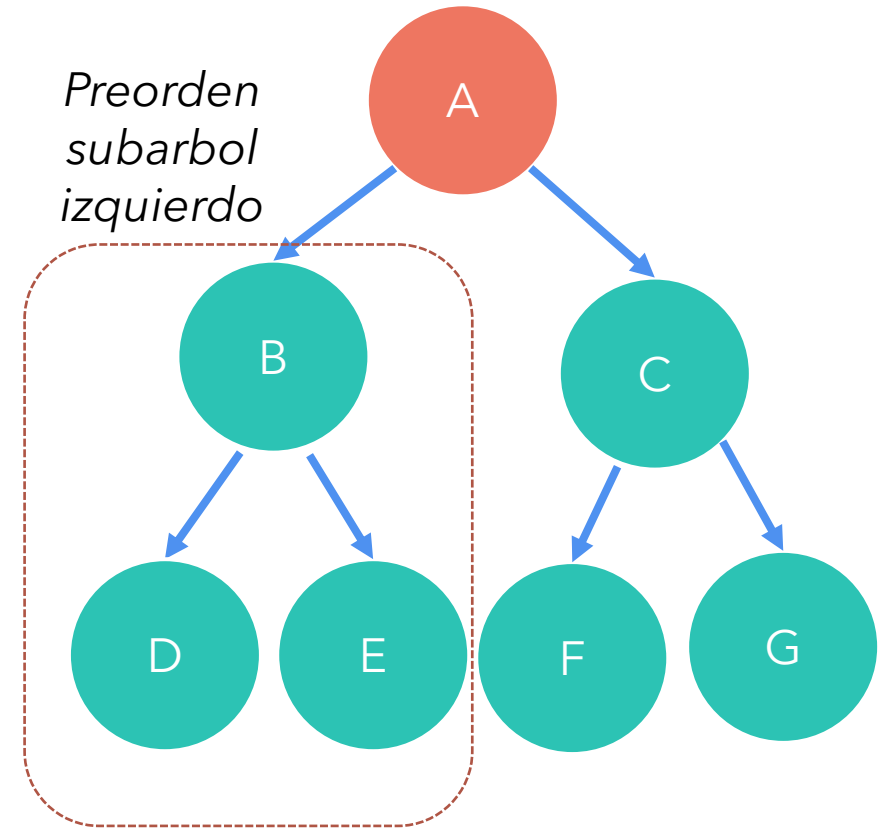
Preorden(T) = A -



Arboles – Algoritmos de Recorrido

Algoritmo pre-orden:

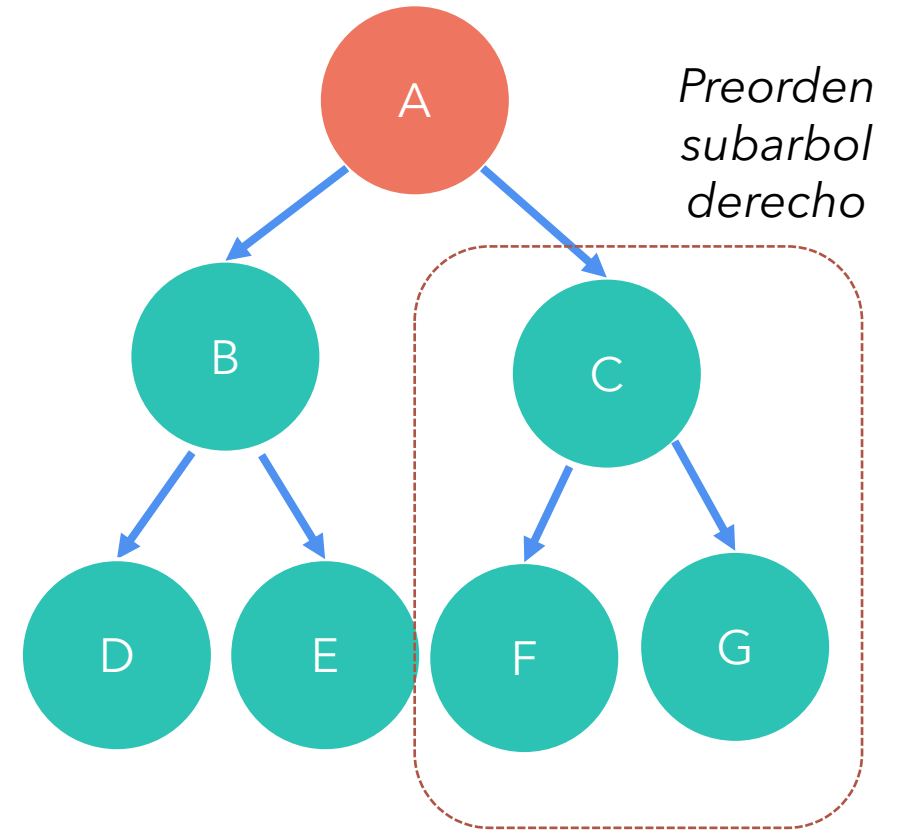
Preorden(T) = A -



Arboles – Algoritmos de Recorrido

Algoritmo pre-orden:

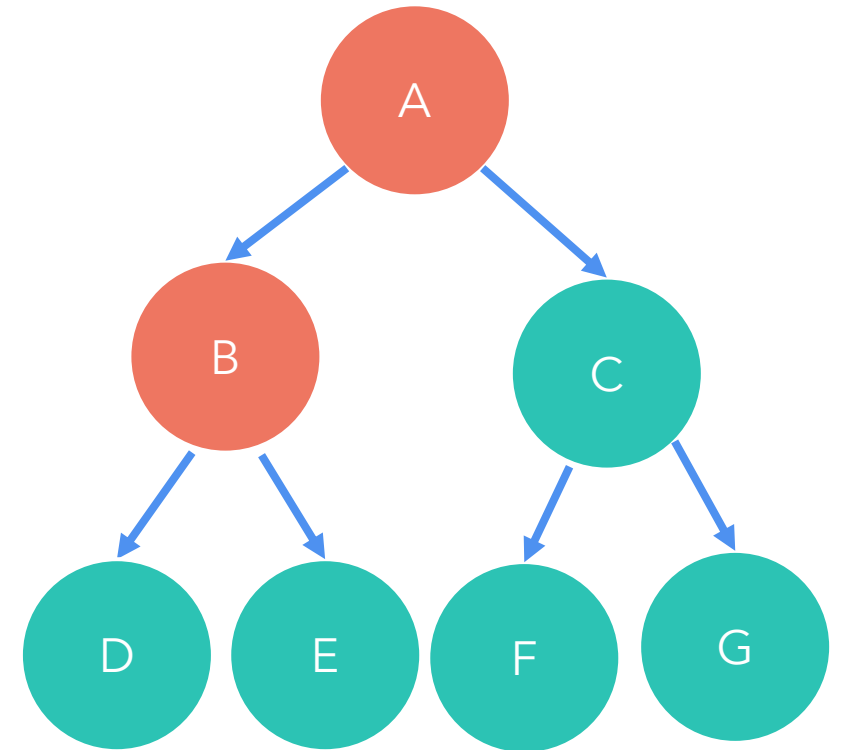
Preorden(T) = A -



Arboles – Algoritmos de Recorrido

Algoritmo pre-orden:

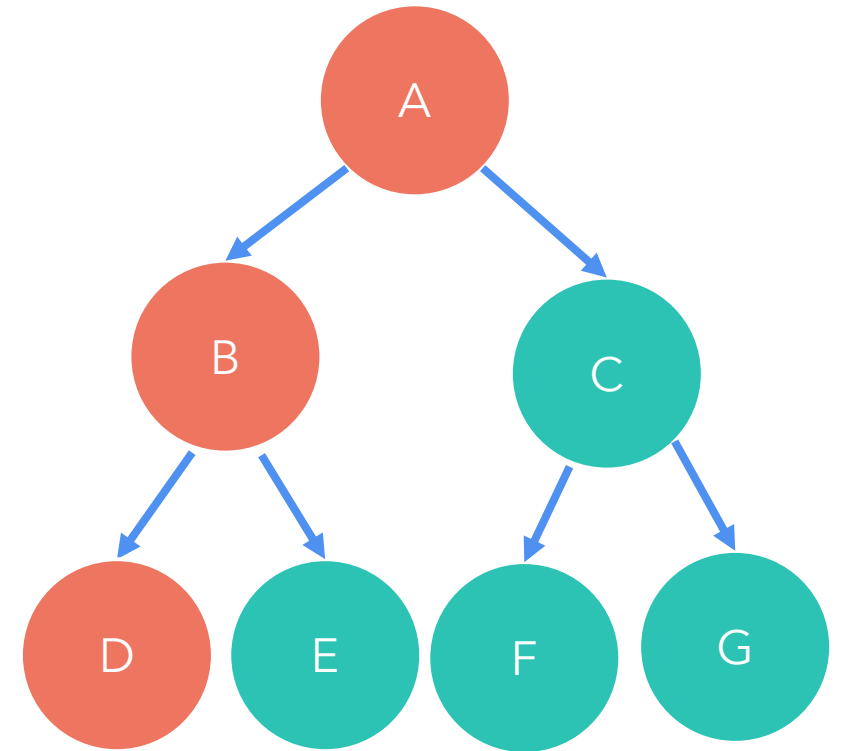
Preorden(T) = A – B



Arboles – Algoritmos de Recorrido

Algoritmo pre-orden:

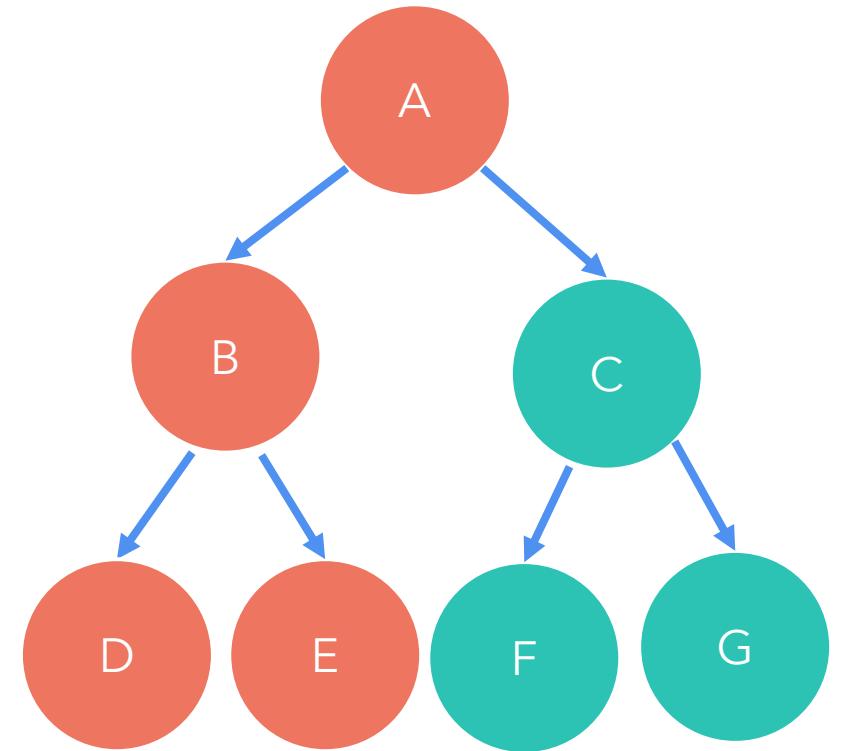
Preorden(T) = A – B – D



Arboles – Algoritmos de Recorrido

Algoritmo pre-orden:

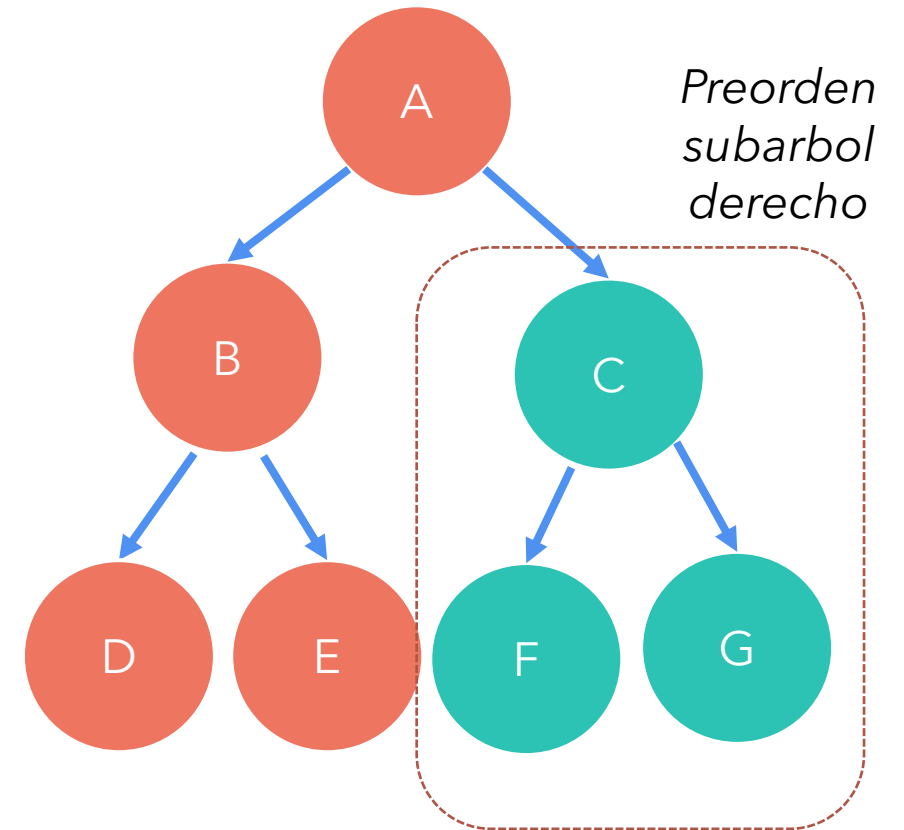
Preorden(T) = A – B – D – E



Arboles – Algoritmos de Recorrido

Algoritmo pre-orden:

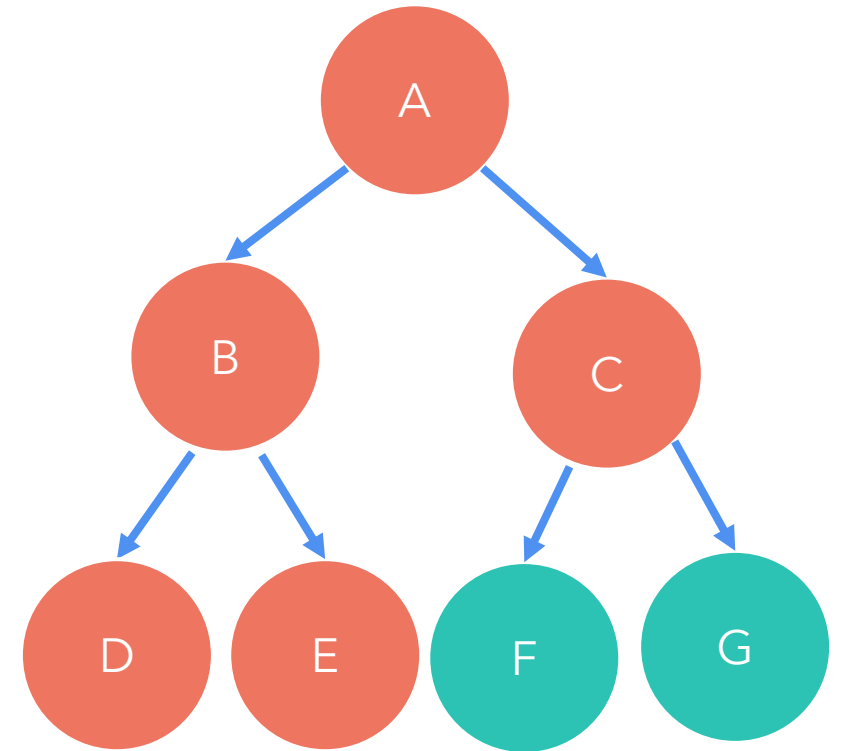
Preorden(T) = A – B – D – E



Arboles – Algoritmos de Recorrido

Algoritmo pre-orden:

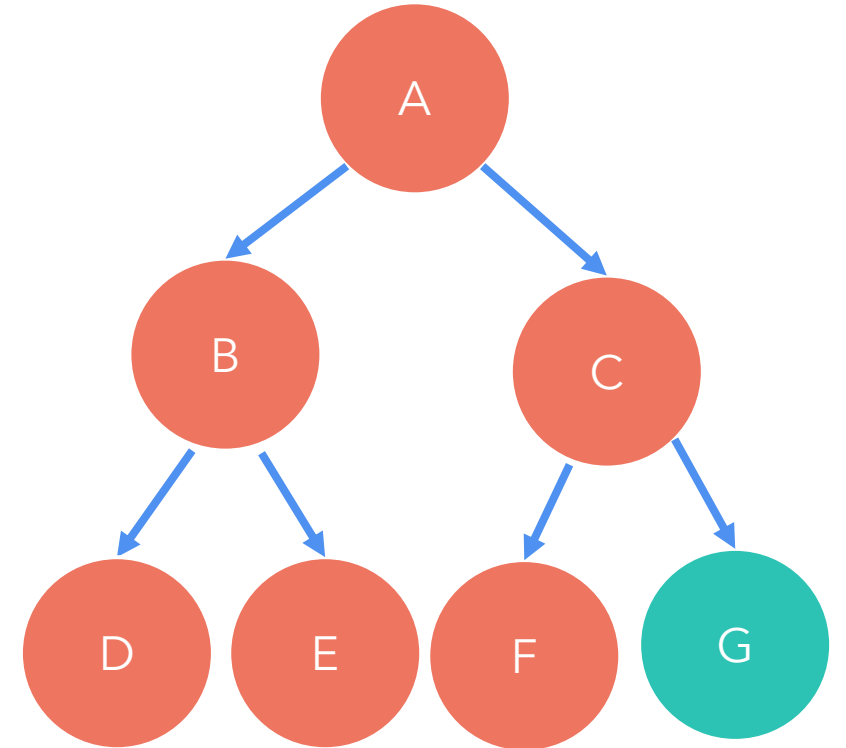
Preorden(T) = A – B – D – E – C



Arboles – Algoritmos de Recorrido

Algoritmo pre-orden:

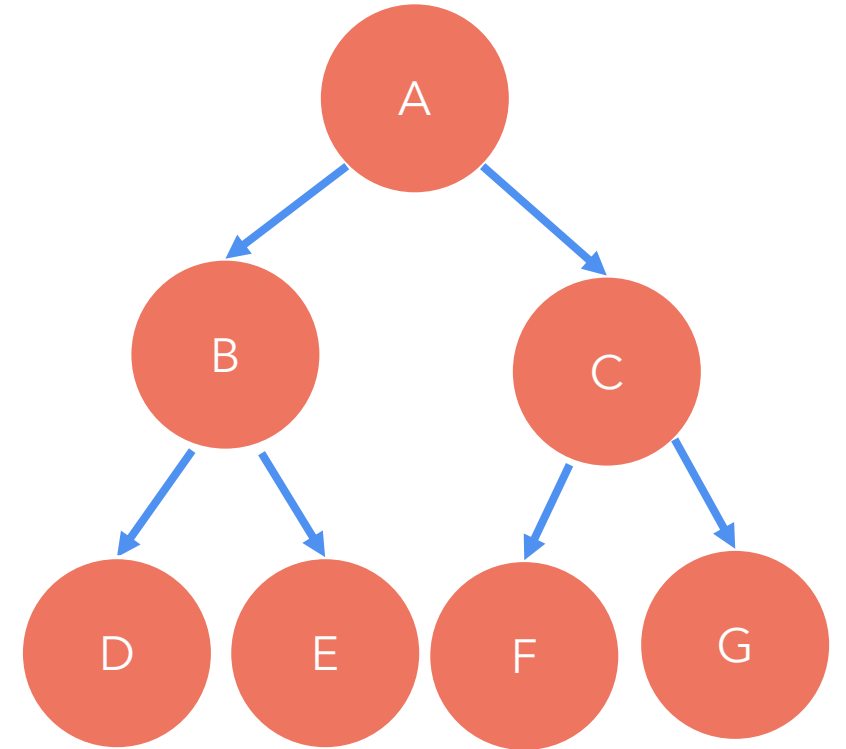
Preorden(T) = A – B – D – E – C – F



Arboles – Algoritmos de Recorrido

Algoritmo pre-orden:

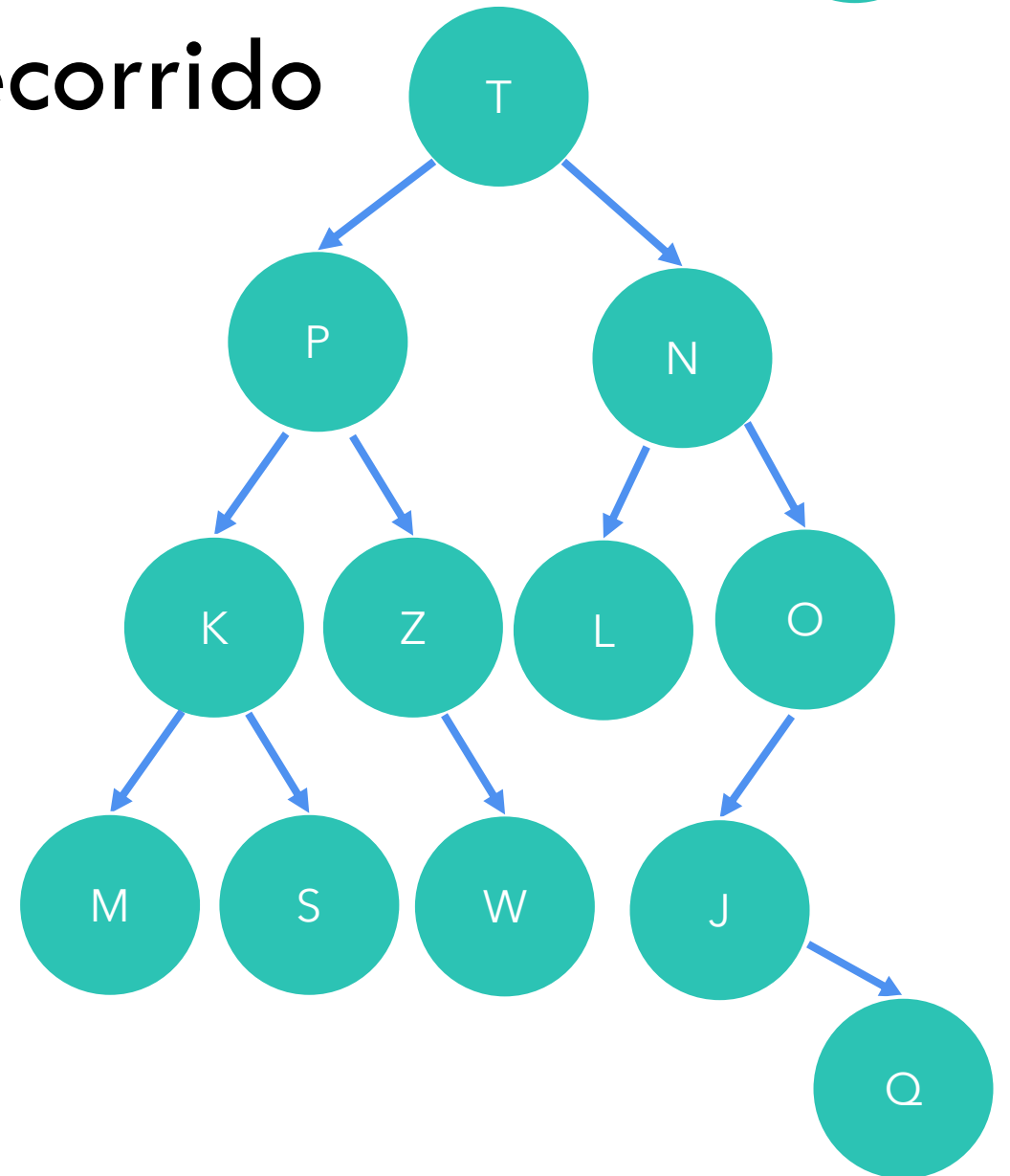
Preorden(T) = A – B – D – E – C – F – G



Arboles – Algoritmos de Recorrido

Algoritmo pre-orden:

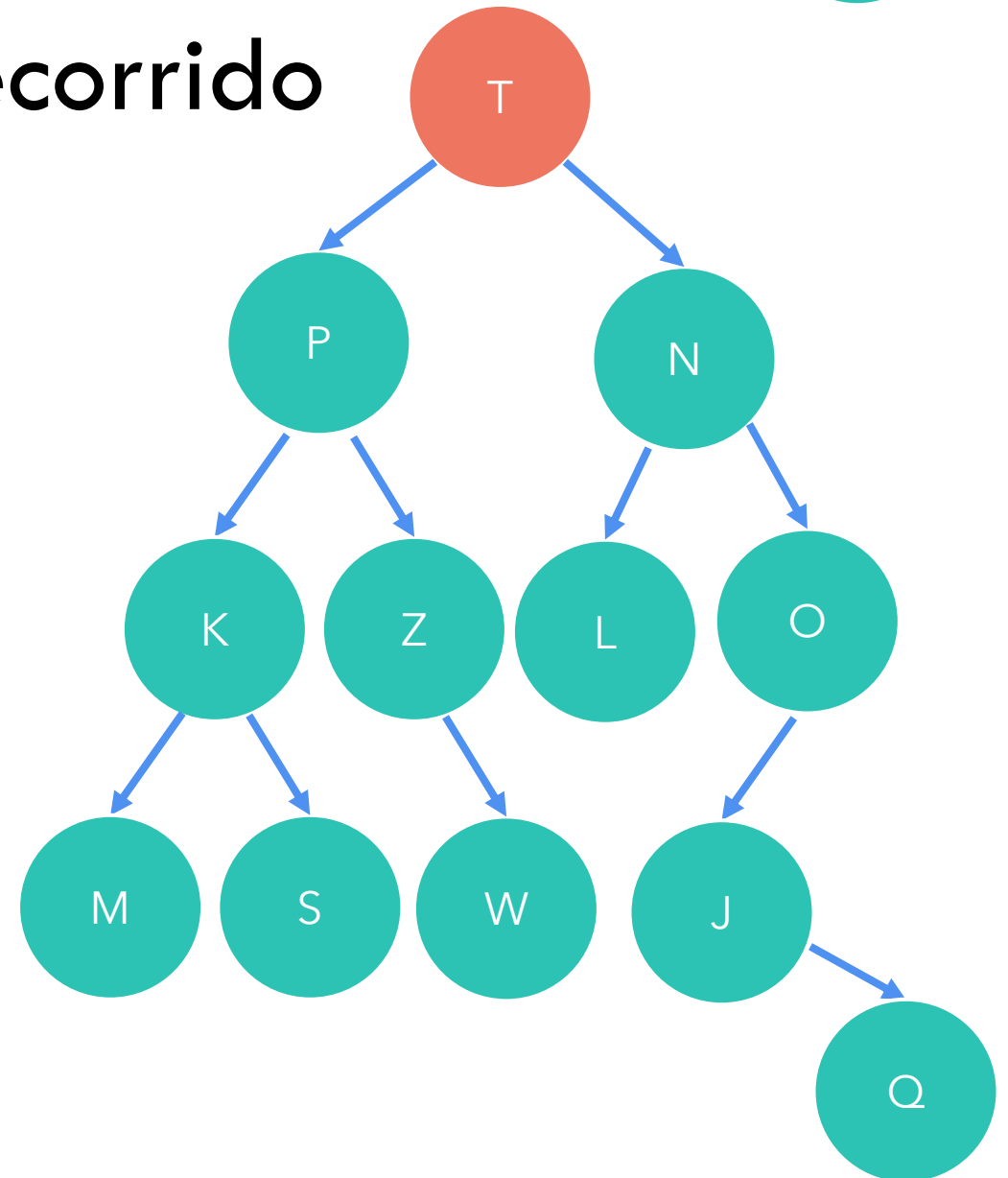
Preorden(T) =



Arboles – Algoritmos de Recorrido

Algoritmo pre-orden:

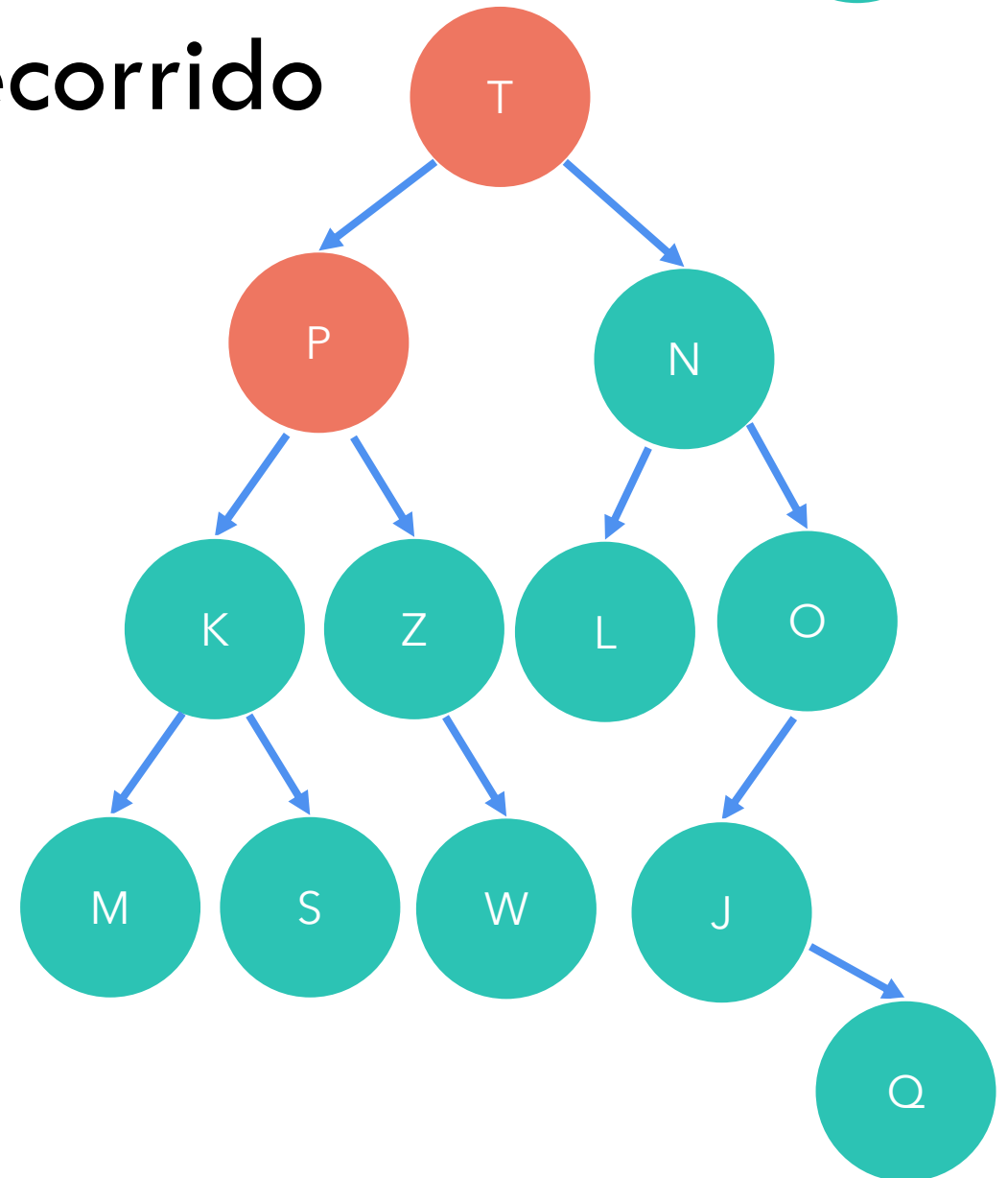
Preorden(T) = T -



Arboles – Algoritmos de Recorrido

Algoritmo pre-orden:

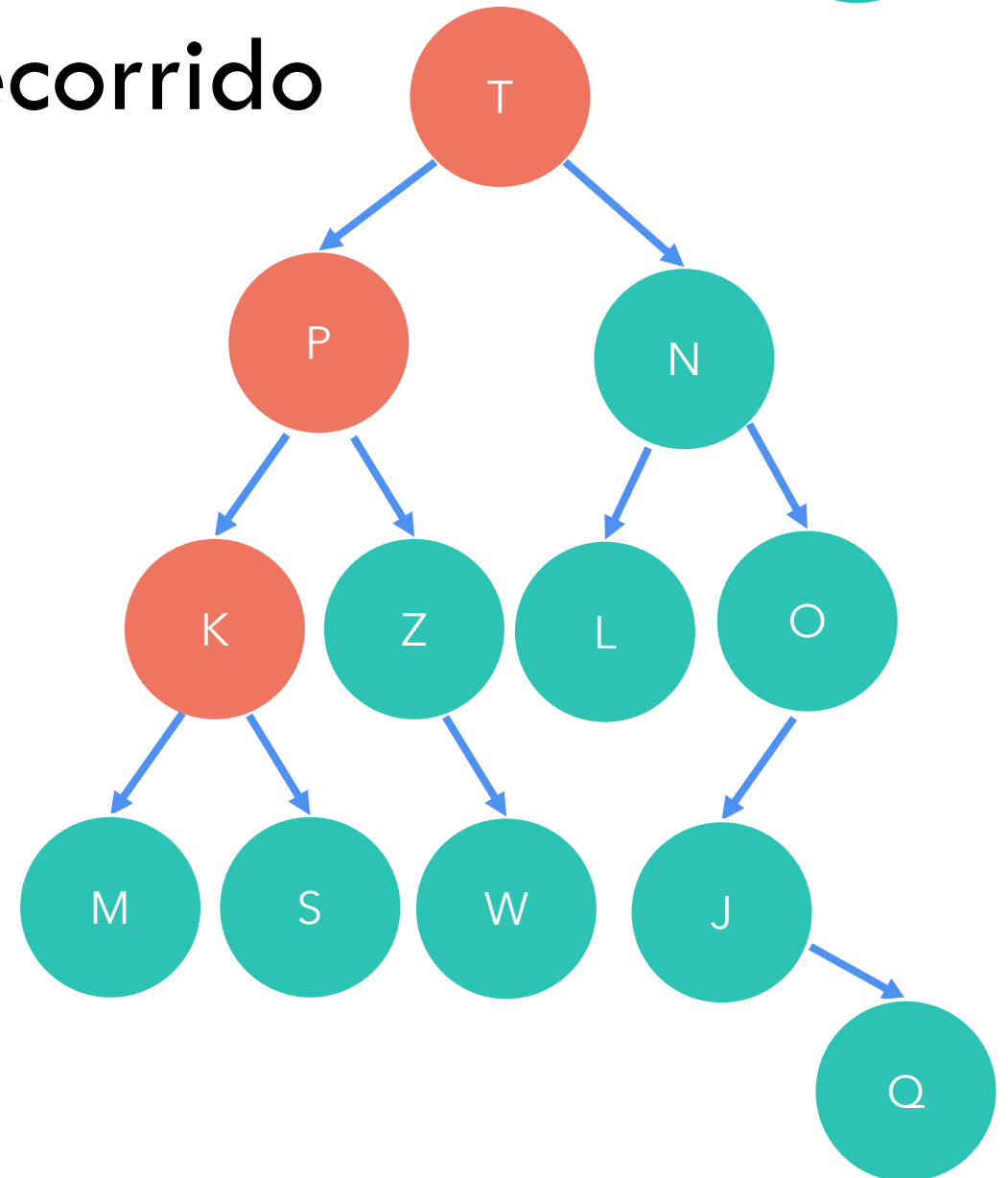
$\text{Preorden}(T) = T - P$



Arboles – Algoritmos de Recorrido

Algoritmo pre-orden:

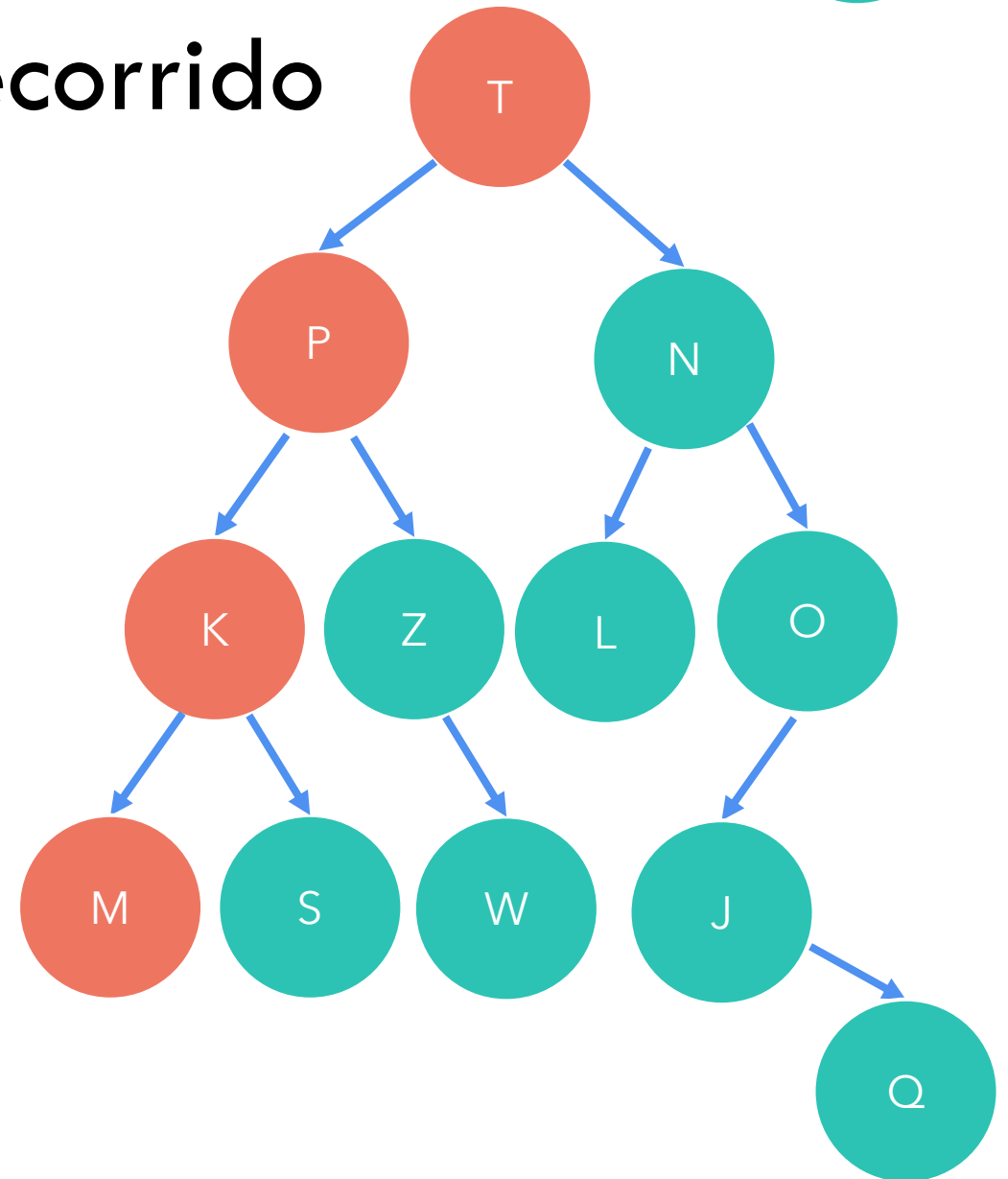
Preorden(T) = T – P – K



Arboles – Algoritmos de Recorrido

Algoritmo pre-orden:

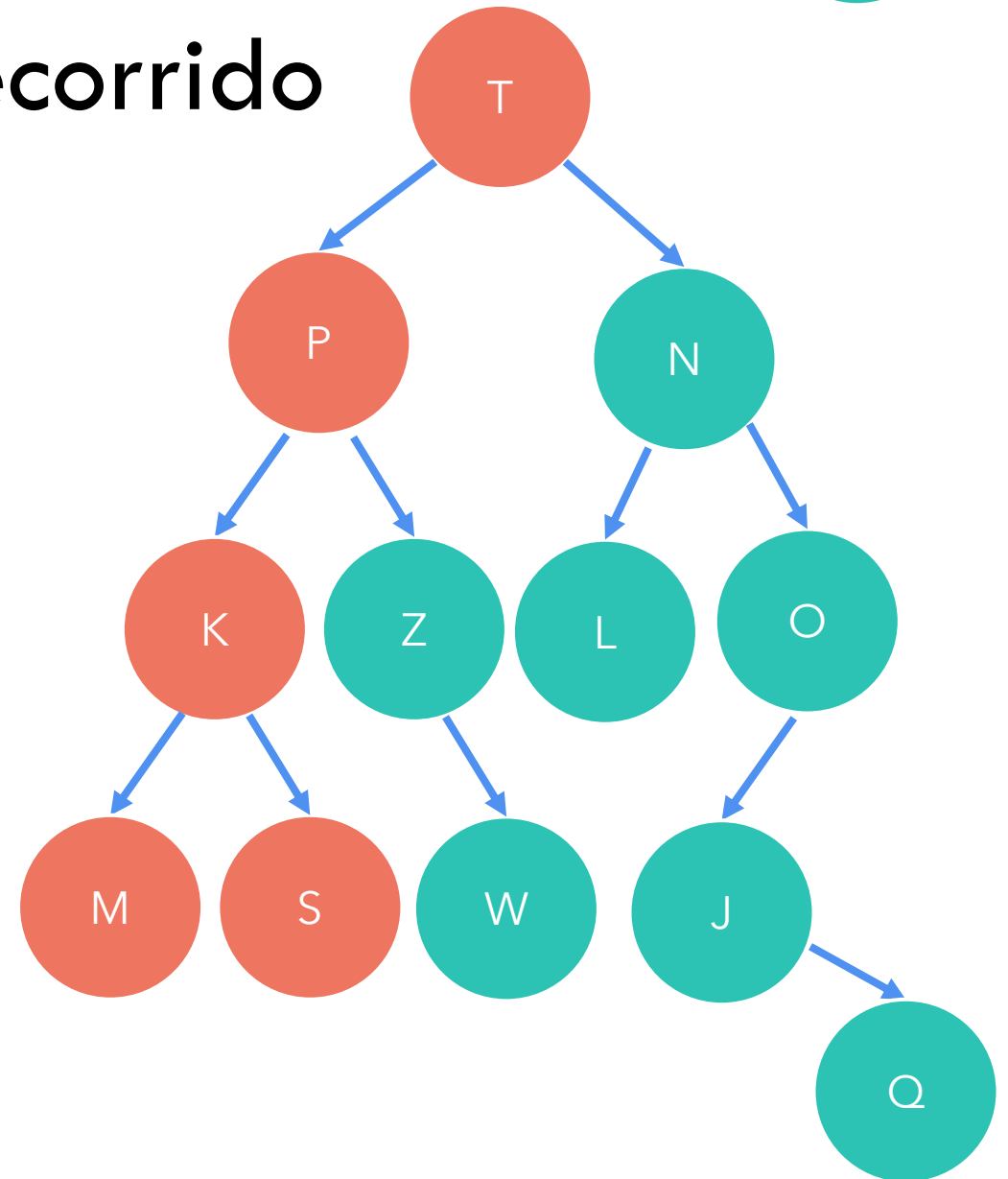
Preorden(T) = T – P – K – M



Arboles – Algoritmos de Recorrido

Algoritmo pre-orden:

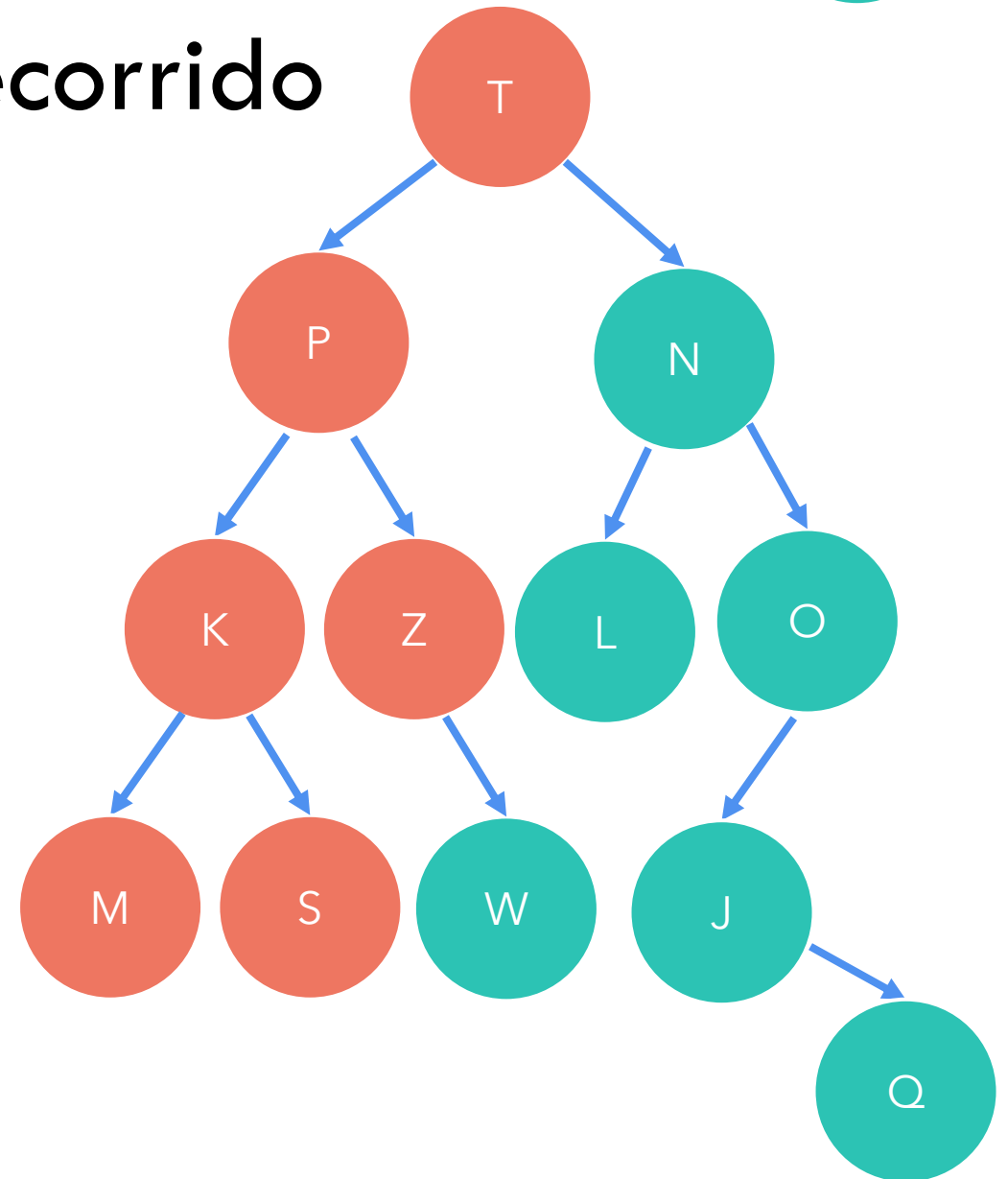
Preorden(T) = T – P – K – M – S



Arboles – Algoritmos de Recorrido

Algoritmo pre-orden:

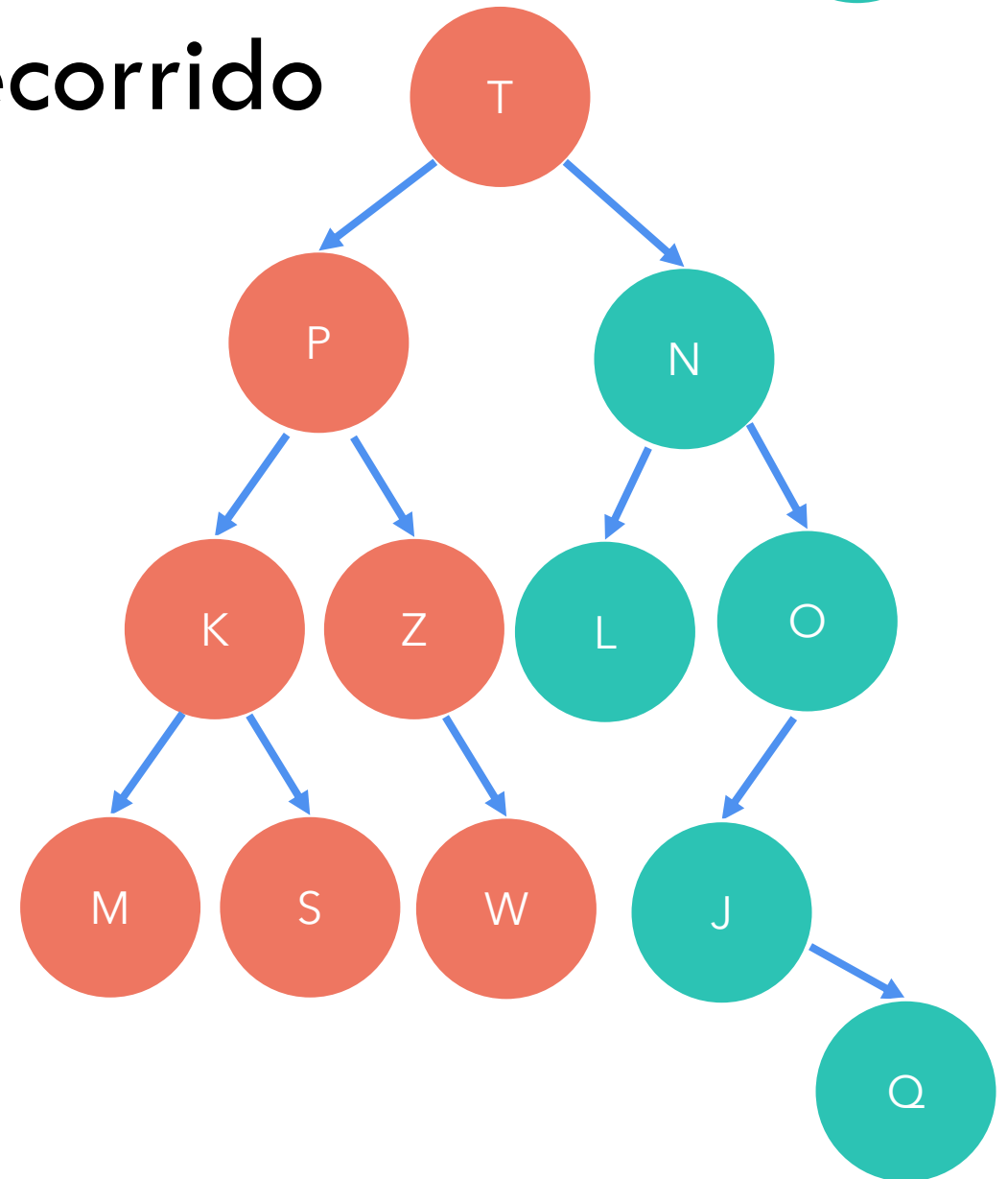
Preorden(T) = T – P – K – M – S – Z



Arboles – Algoritmos de Recorrido

Algoritmo pre-orden:

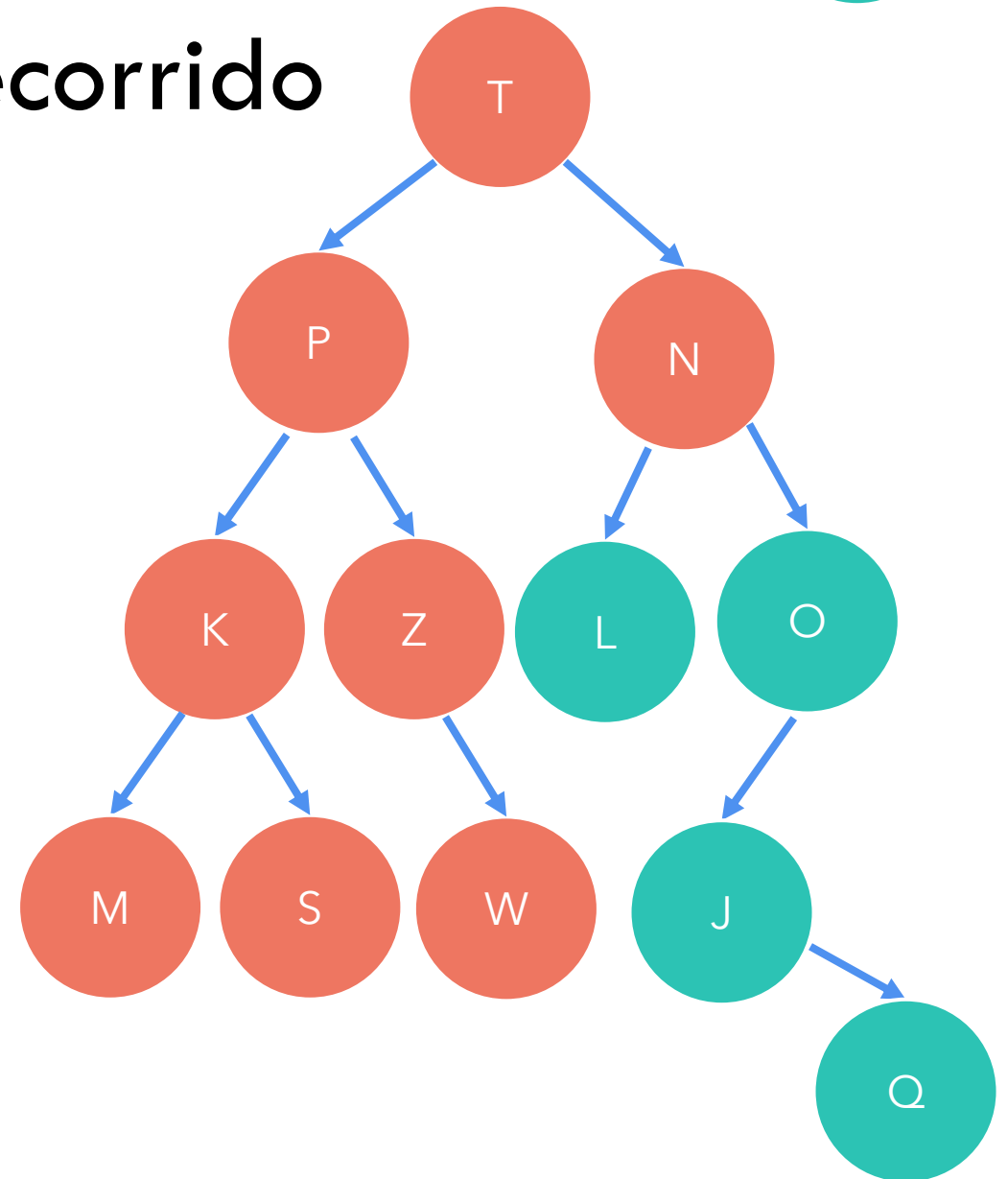
$\text{Preorden}(T) = T - P - K - M - S - Z - W$



Arboles – Algoritmos de Recorrido

Algoritmo pre-orden:

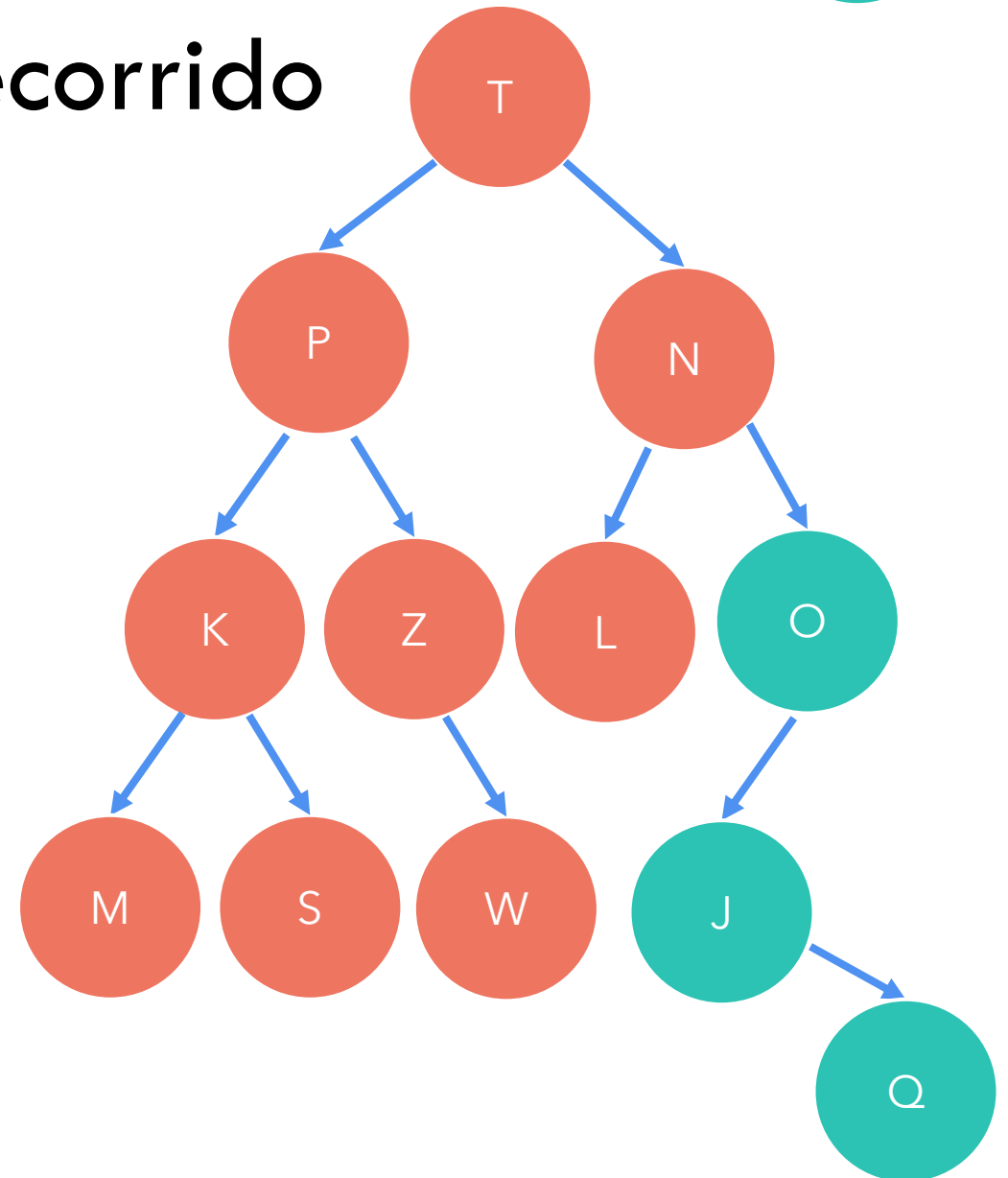
Preorden(T) = T – P – K – M – S – Z – W – N



Arboles – Algoritmos de Recorrido

Algoritmo pre-orden:

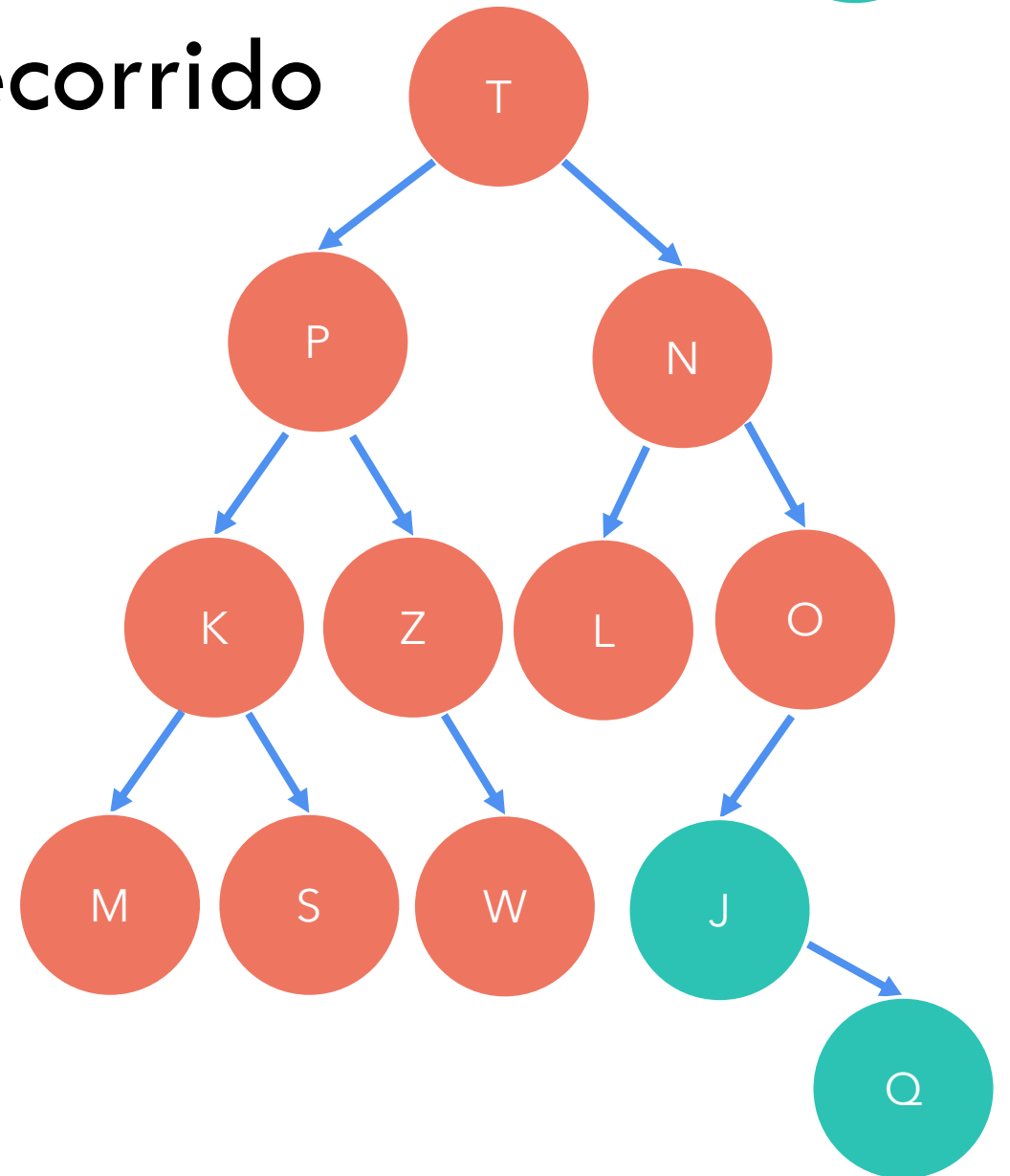
Preorden(T) = T – P – K – M – S – Z – W – N – L



Arboles – Algoritmos de Recorrido

Algoritmo pre-orden:

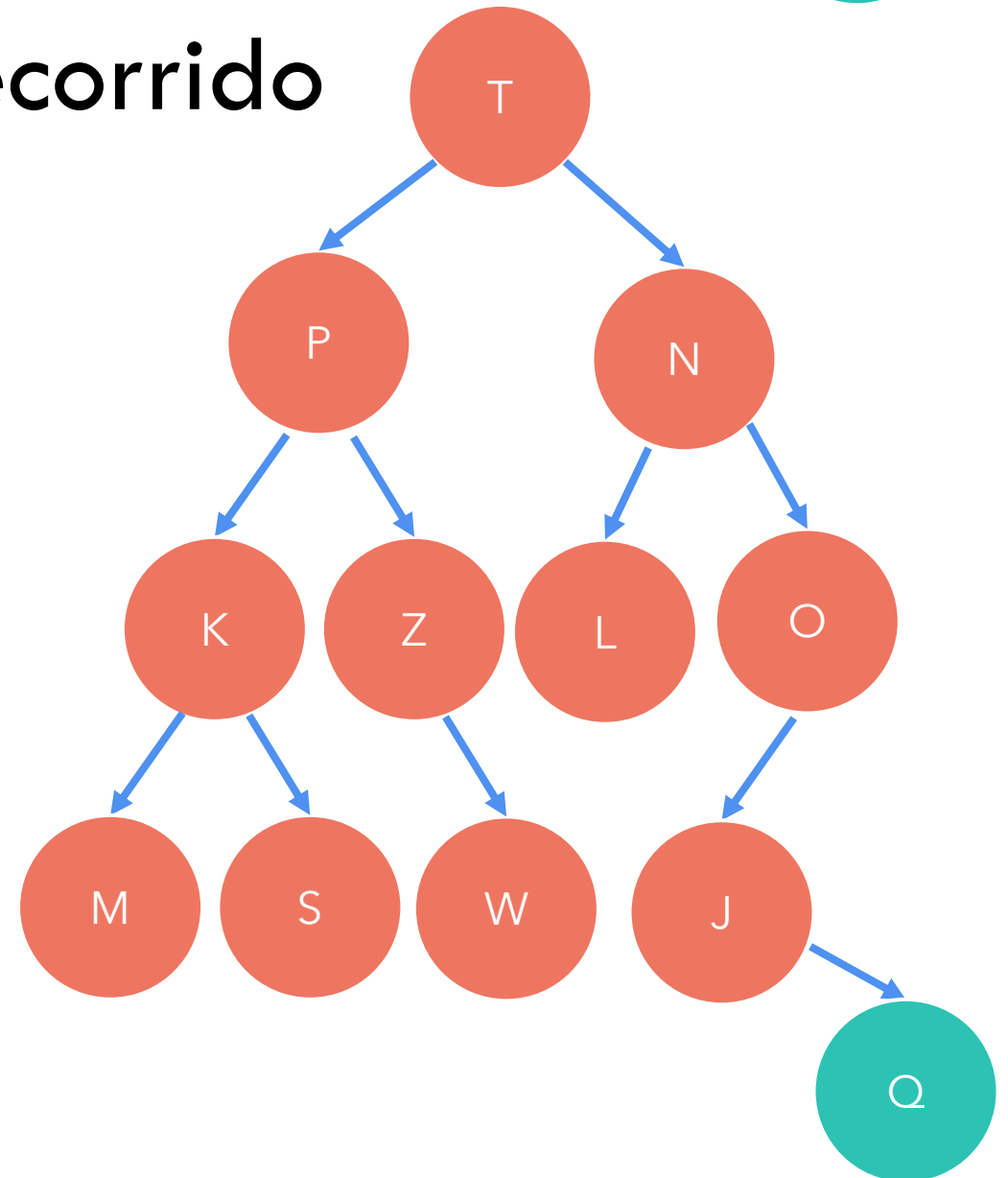
Preorden(T) = T – P – K – M – S – Z – W – N – L – O



Arboles – Algoritmos de Recorrido

Algoritmo pre-orden:

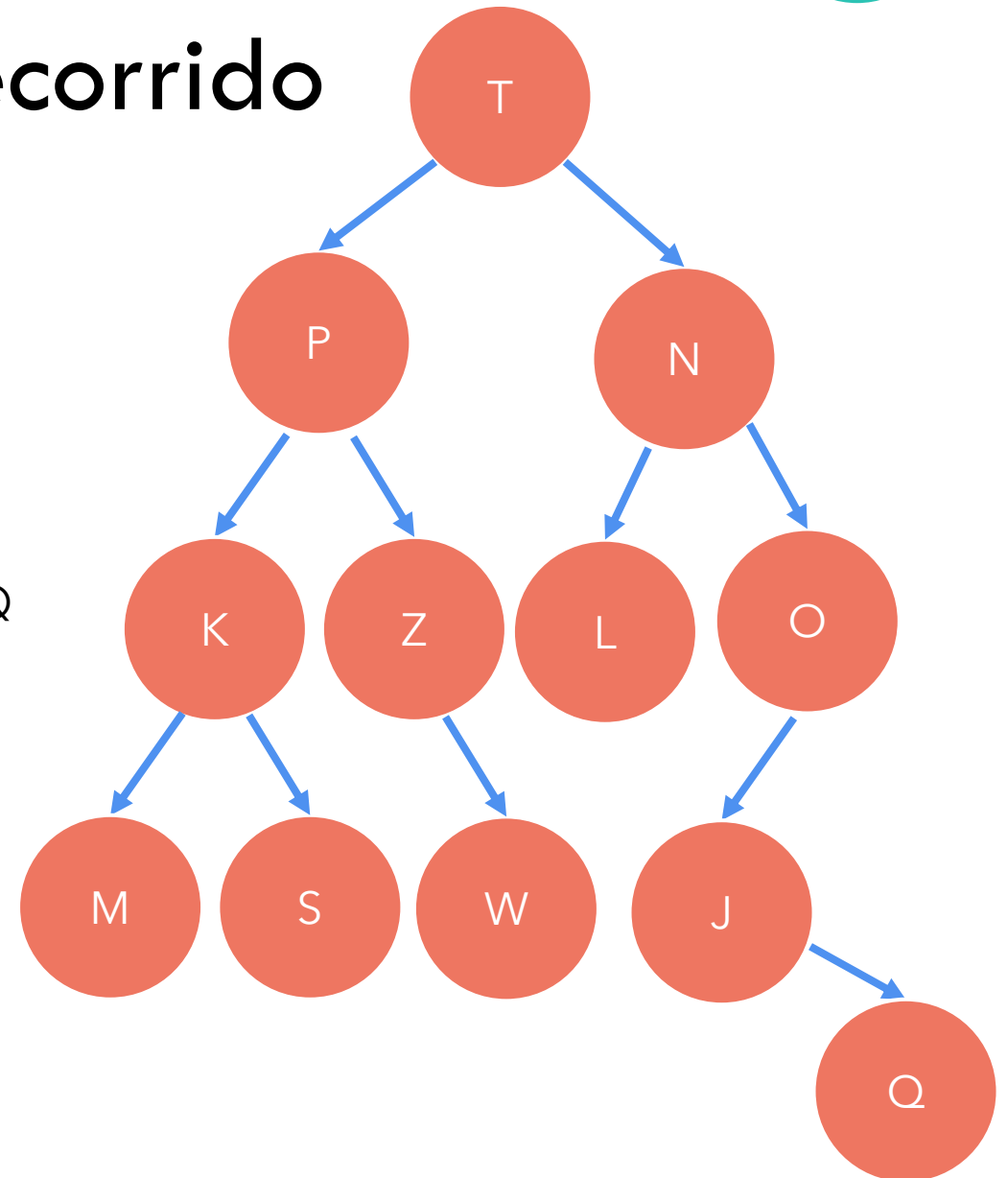
Preorden(T) = T – P – K – M – S – Z – W – N – L – O – J



Arboles – Algoritmos de Recorrido

Algoritmo pre-orden:

Preorden(T) = T – P – K – M – S – Z – W – N – L – O – J – Q



Arboles – Algoritmos de Recorrido

Algoritmo pre-orden:

Seudocodigo

Preorder(BinaryTree T, Node v)

1. visit(v)
2. if T.hasLeft(v)
3. Preorden(T,T.left(v))
4. if T.hasRight(v)
5. Preorden(T,T.right(v))

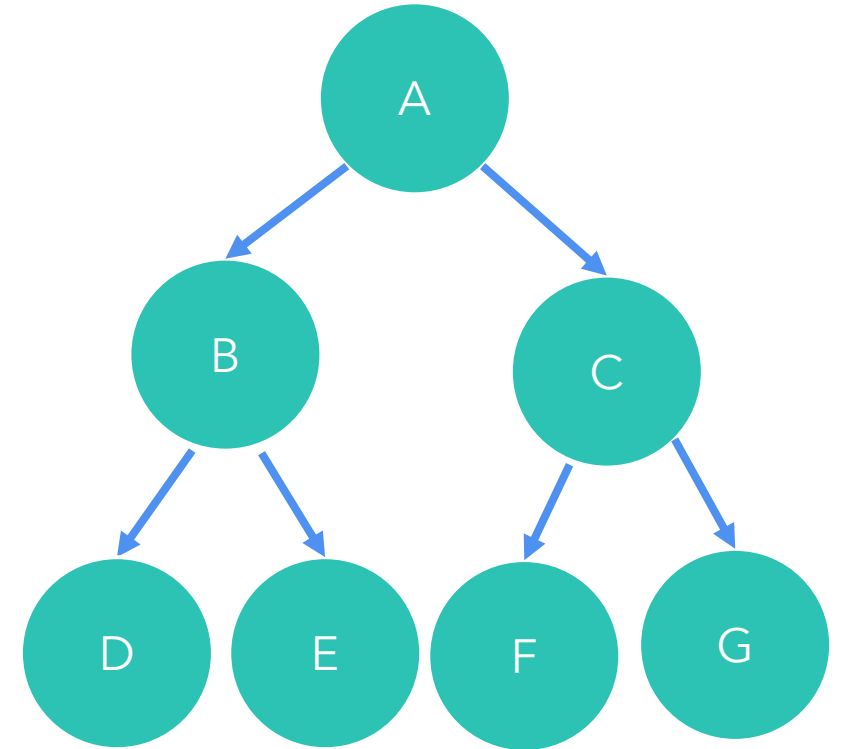
El método visitar se puede modificar de acuerdo con el problema. Ejemplos de operación visitar son:

- Un método para imprimir o guardar en un archivo el contenido del nodo
- Acumular los valores numéricos almacenados en el nodo

Arboles – Algoritmos de Recorrido

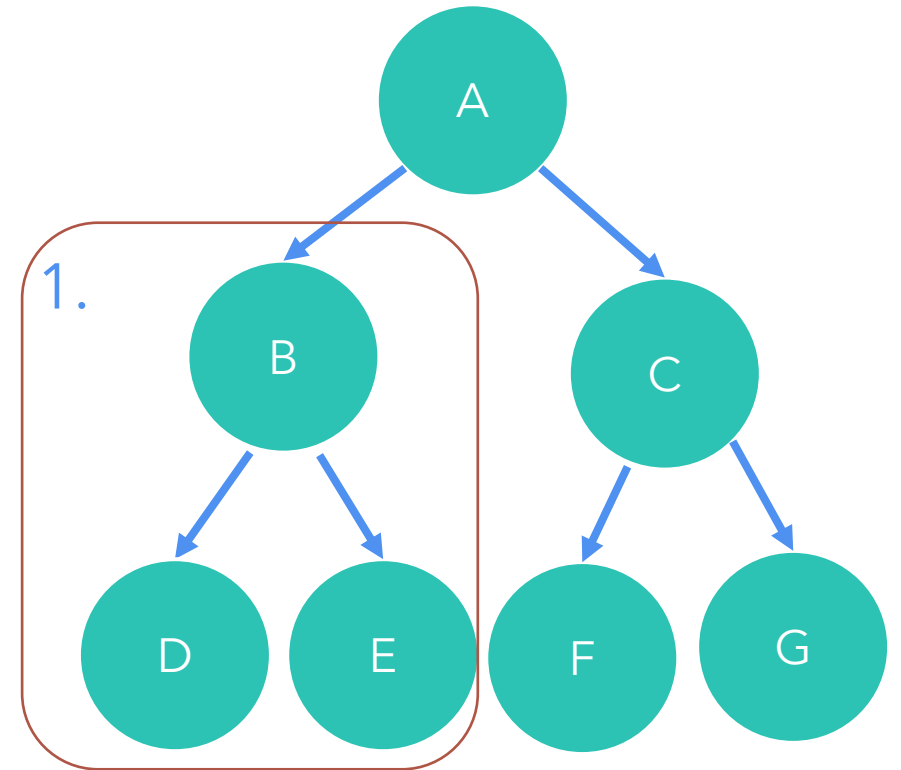
Algoritmo in-orden:

- ❑ En este recorrido primero se visita de forma recursiva el subárbol izquierdo
- ❑ Luego, se visita la raíz
- ❑ Posteriormente, se visita recursivamente el subárbol derecho



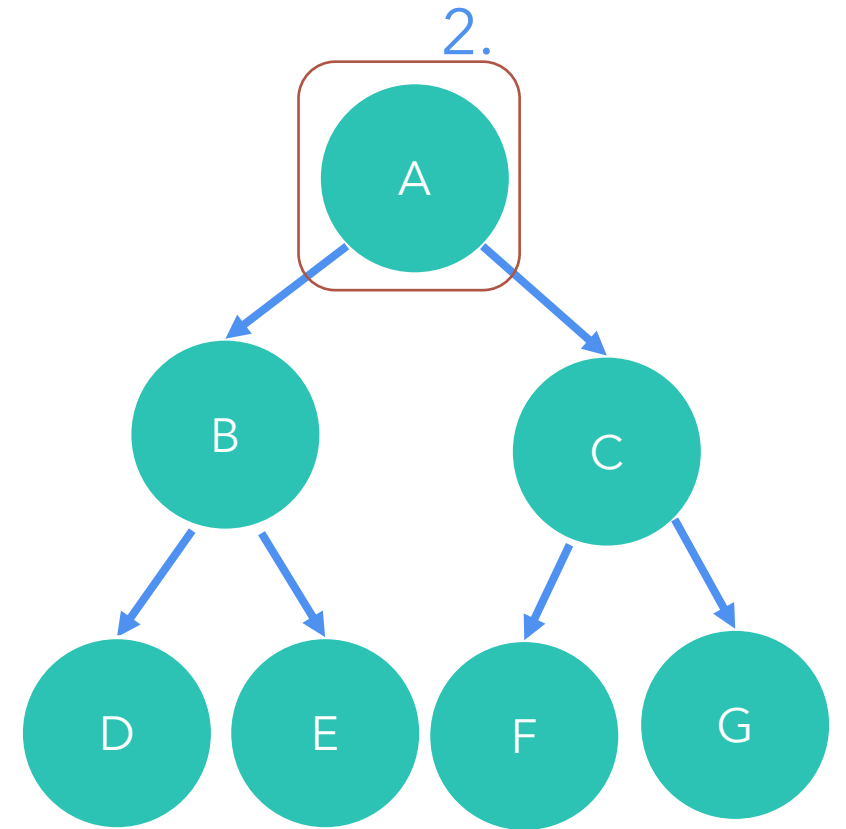
Arboles – Algoritmos de Recorrido

Algoritmo in-orden:



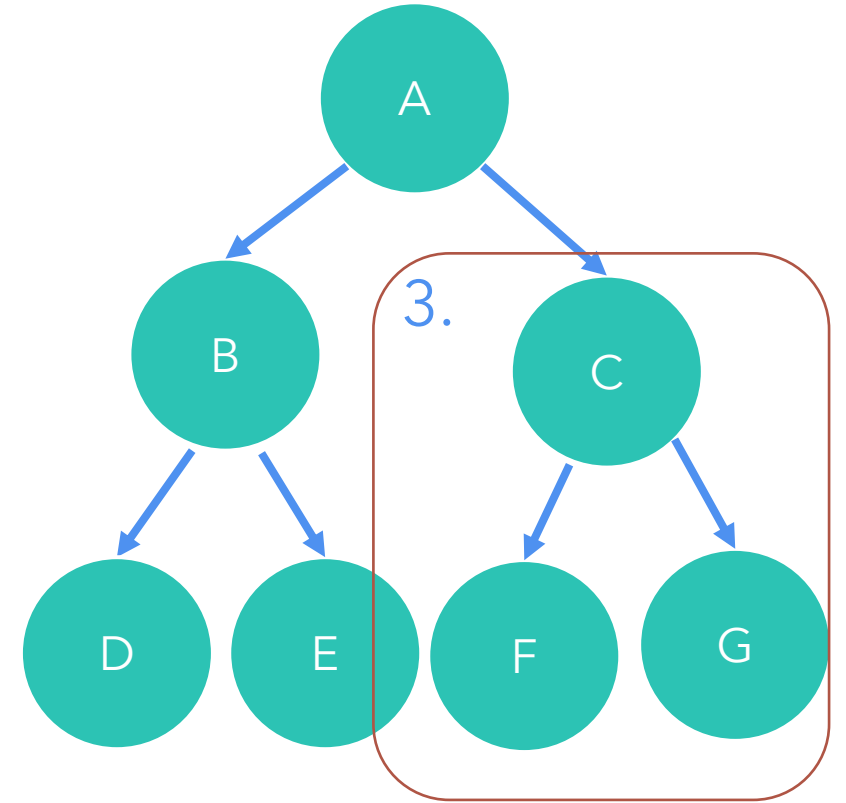
Arboles – Algoritmos de Recorrido

Algoritmo in-orden:



Arboles – Algoritmos de Recorrido

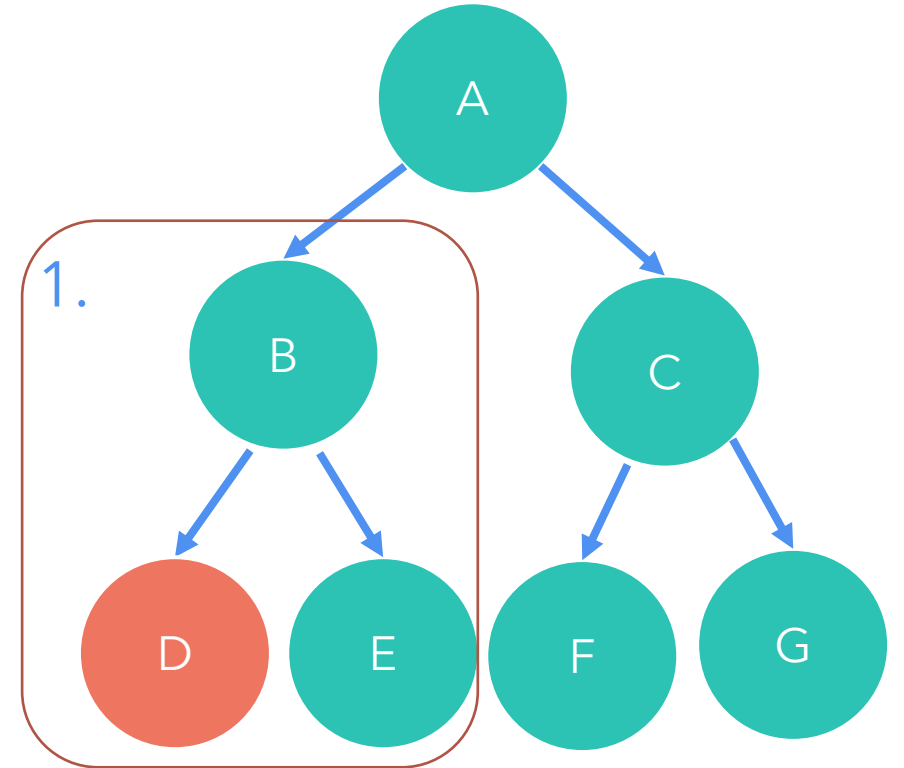
Algoritmo in-orden:



Arboles – Algoritmos de Recorrido

Algoritmo in-orden:

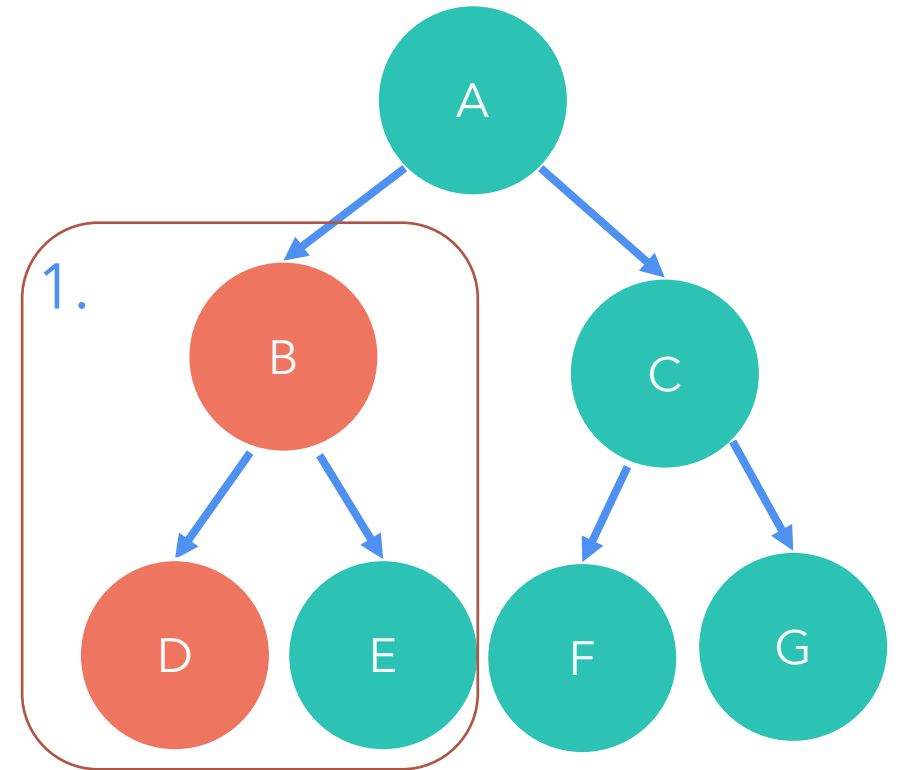
inorder : D-



Arboles – Algoritmos de Recorrido

Algoritmo in-orden:

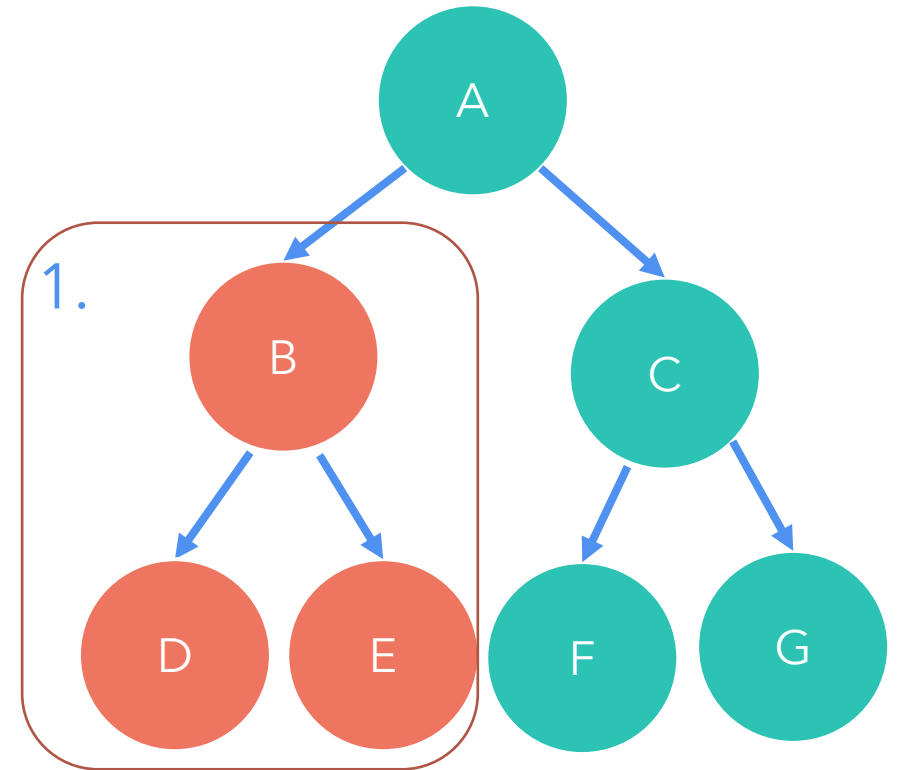
inorder : D-B-



Arboles – Algoritmos de Recorrido

Algoritmo in-orden:

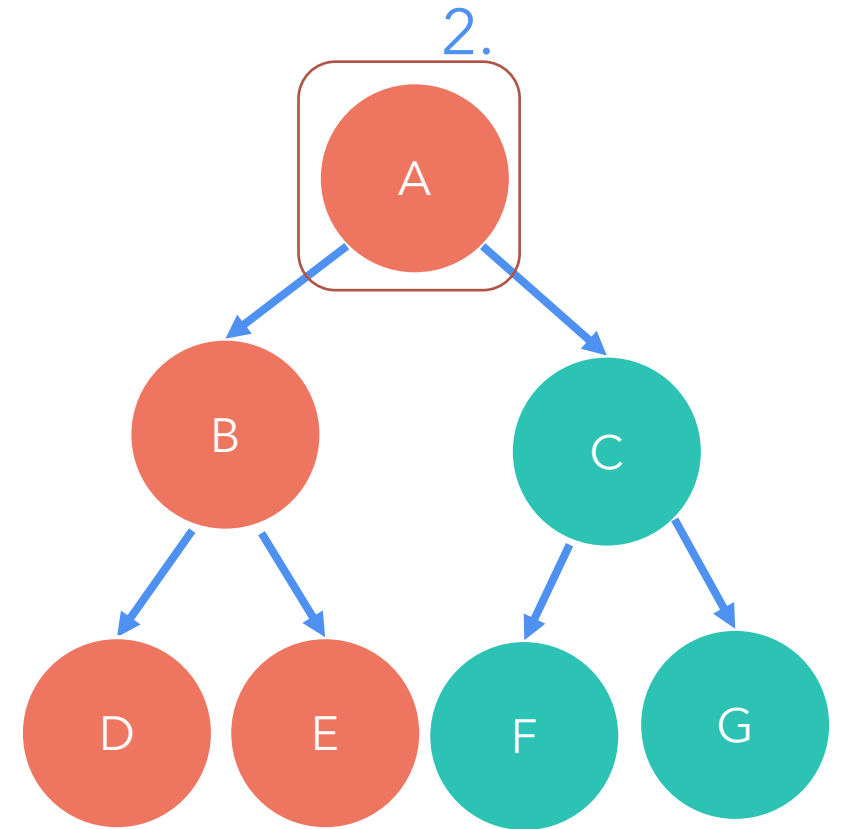
inorder : D-B-E-



Arboles – Algoritmos de Recorrido

Algoritmo in-orden:

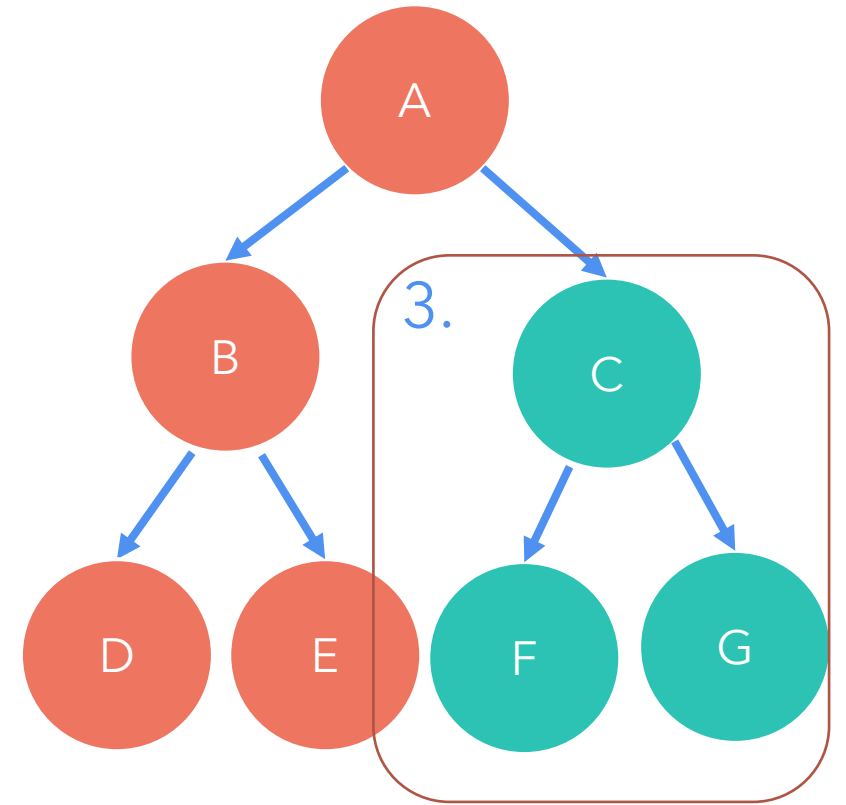
inorder : D-B-E-A-



Arboles – Algoritmos de Recorrido

Algoritmo in-orden:

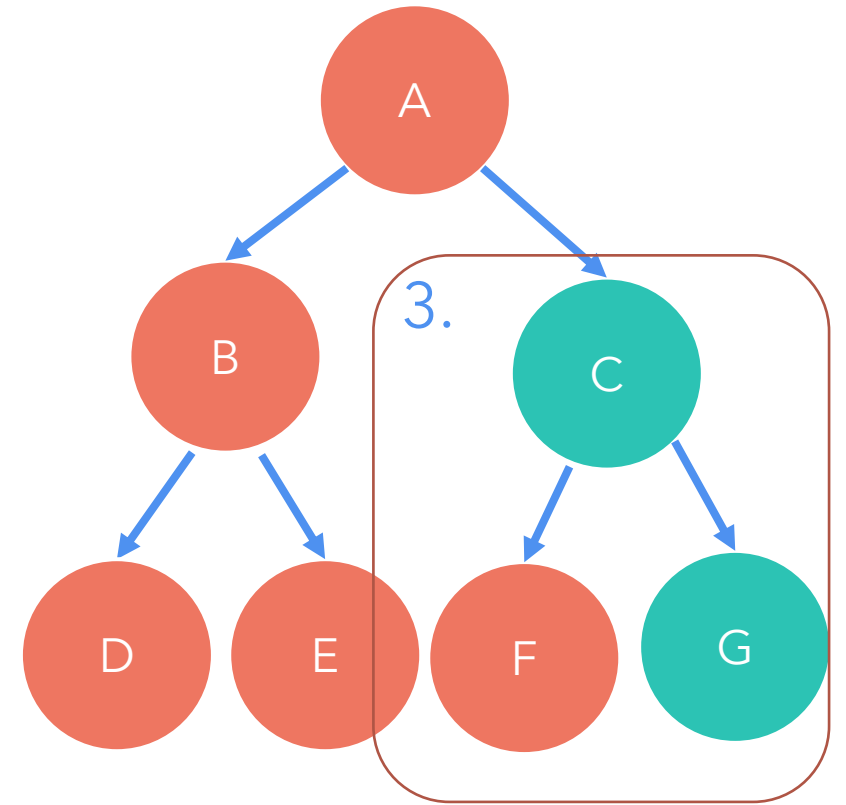
inorder : D-B-E-A-



Arboles – Algoritmos de Recorrido

Algoritmo in-orden:

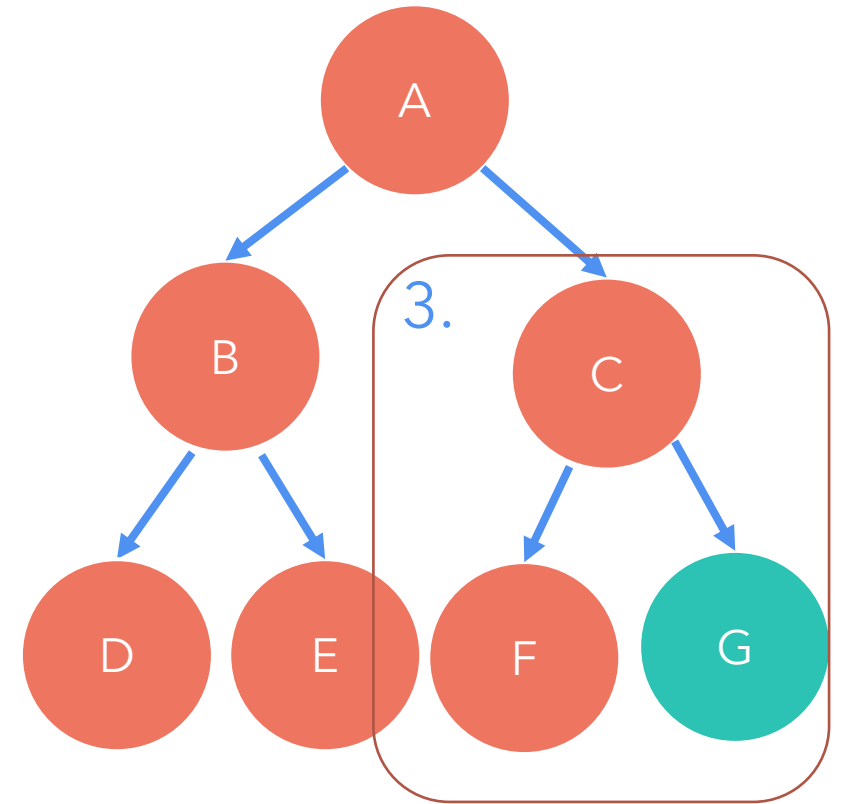
inorder : D-B-E-A-F-



Arboles – Algoritmos de Recorrido

Algoritmo in-orden:

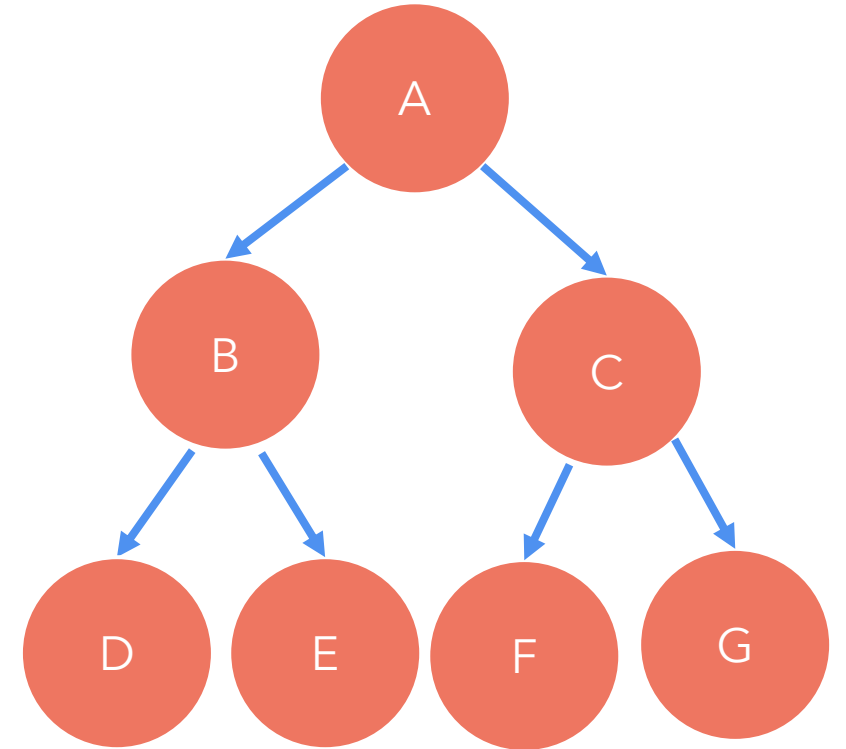
inorder : D-B-E-A-F-C



Arboles – Algoritmos de Recorrido

Algoritmo in-orden:

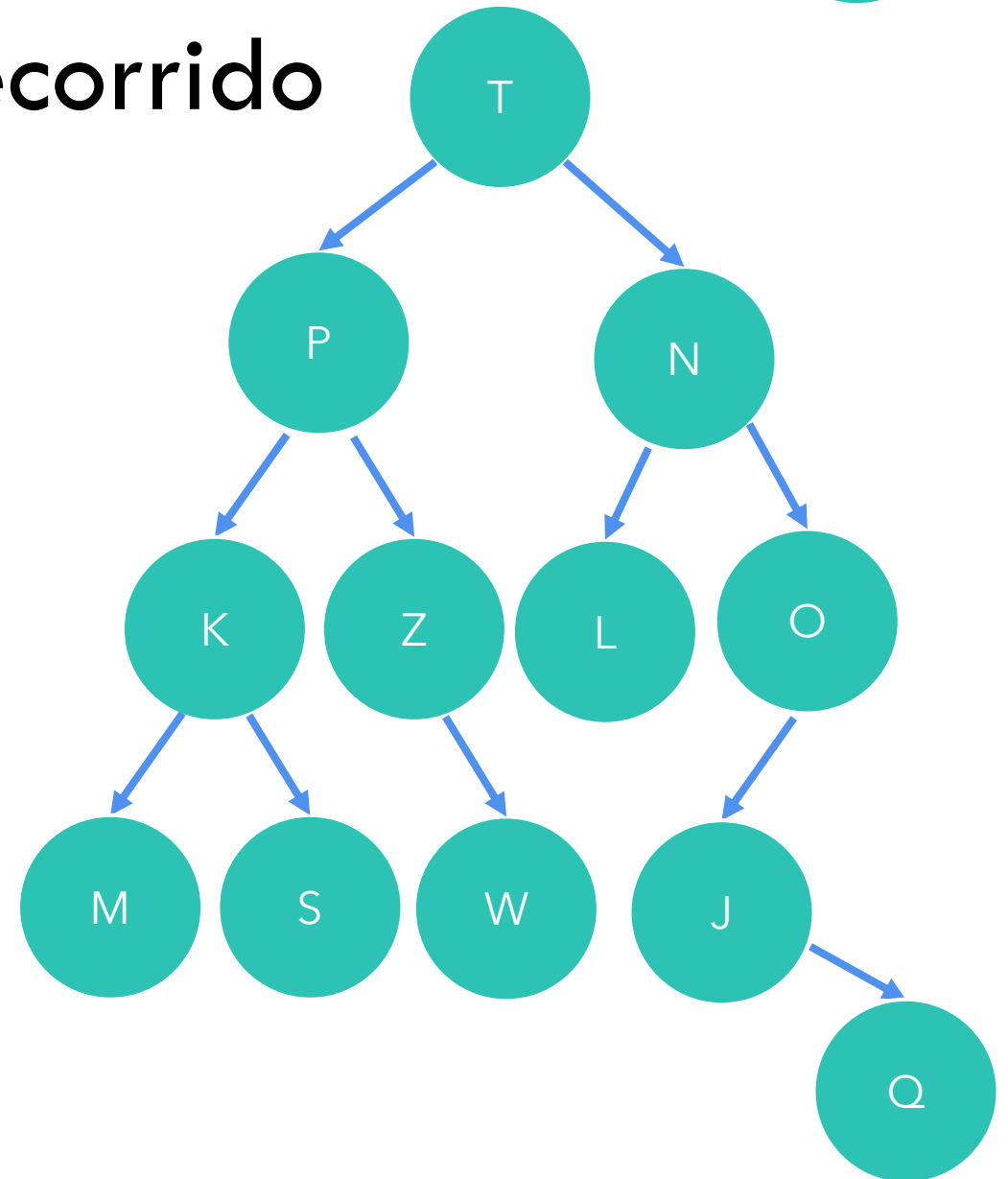
inorder : D-B-E-A-F-C-G



Arboles – Algoritmos de Recorrido

Algoritmo in-orden:

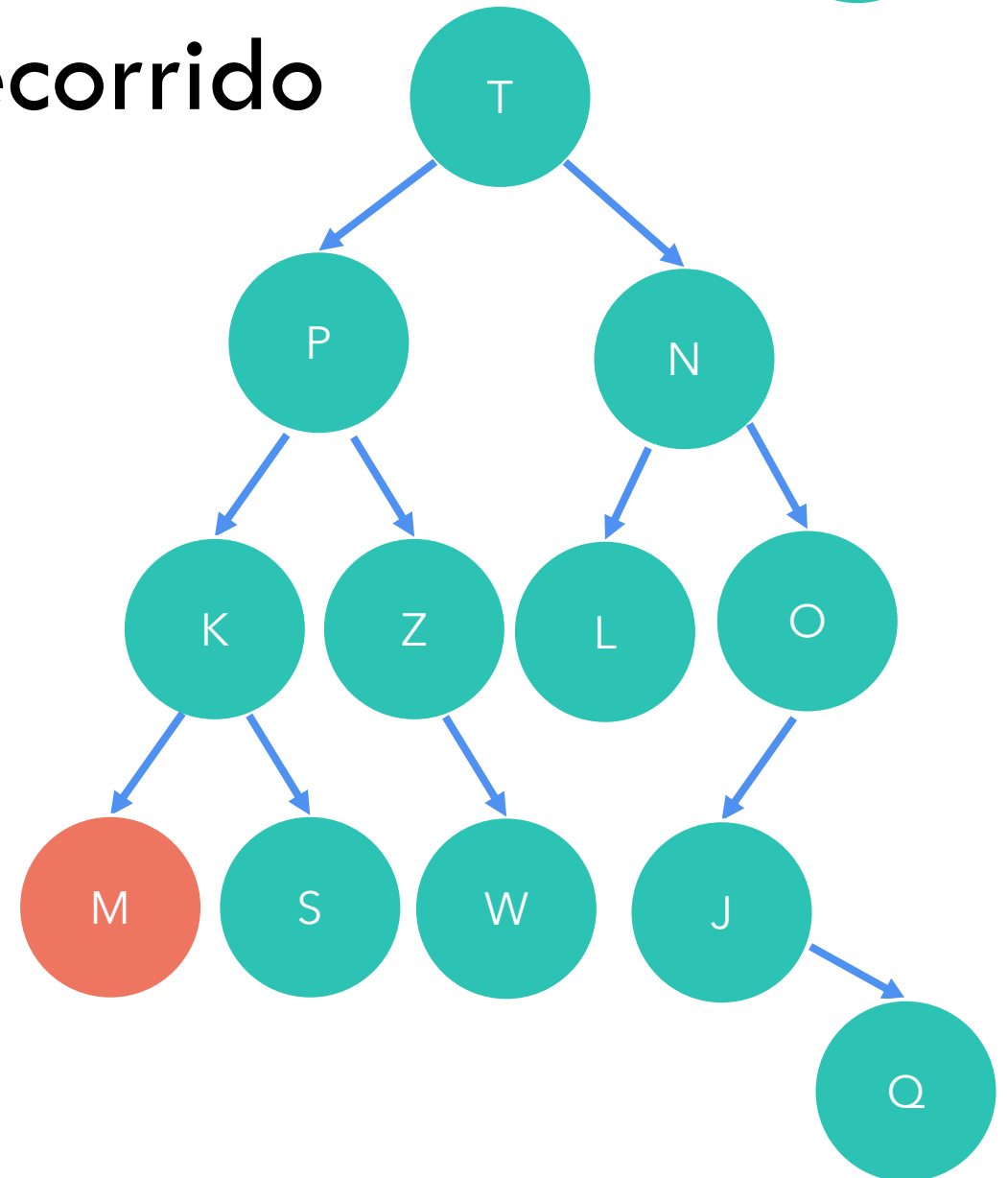
Inorden(T) =



Arboles – Algoritmos de Recorrido

Algoritmo in-orden:

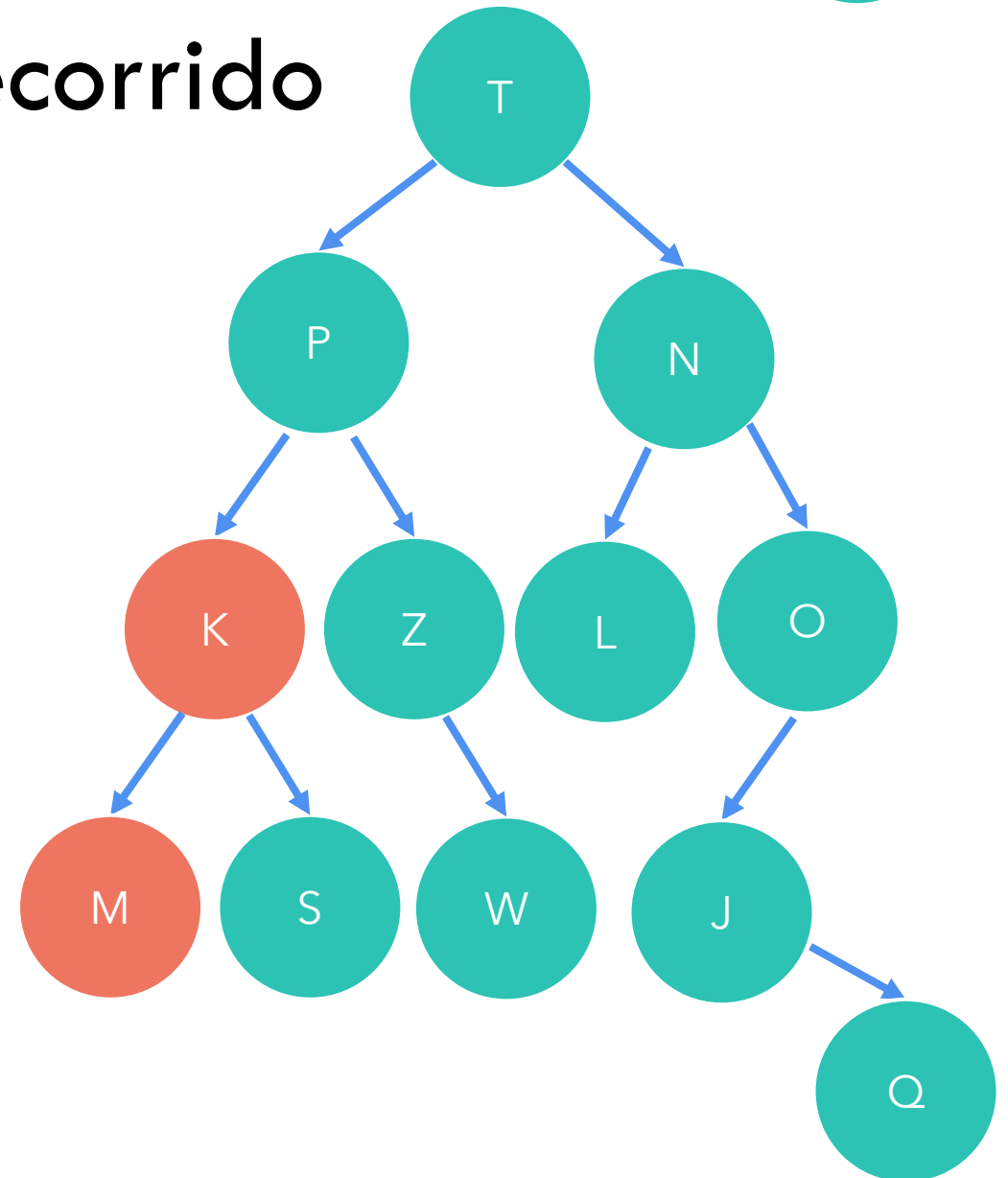
Inorden(T) = M -



Arboles – Algoritmos de Recorrido

Algoritmo in-orden:

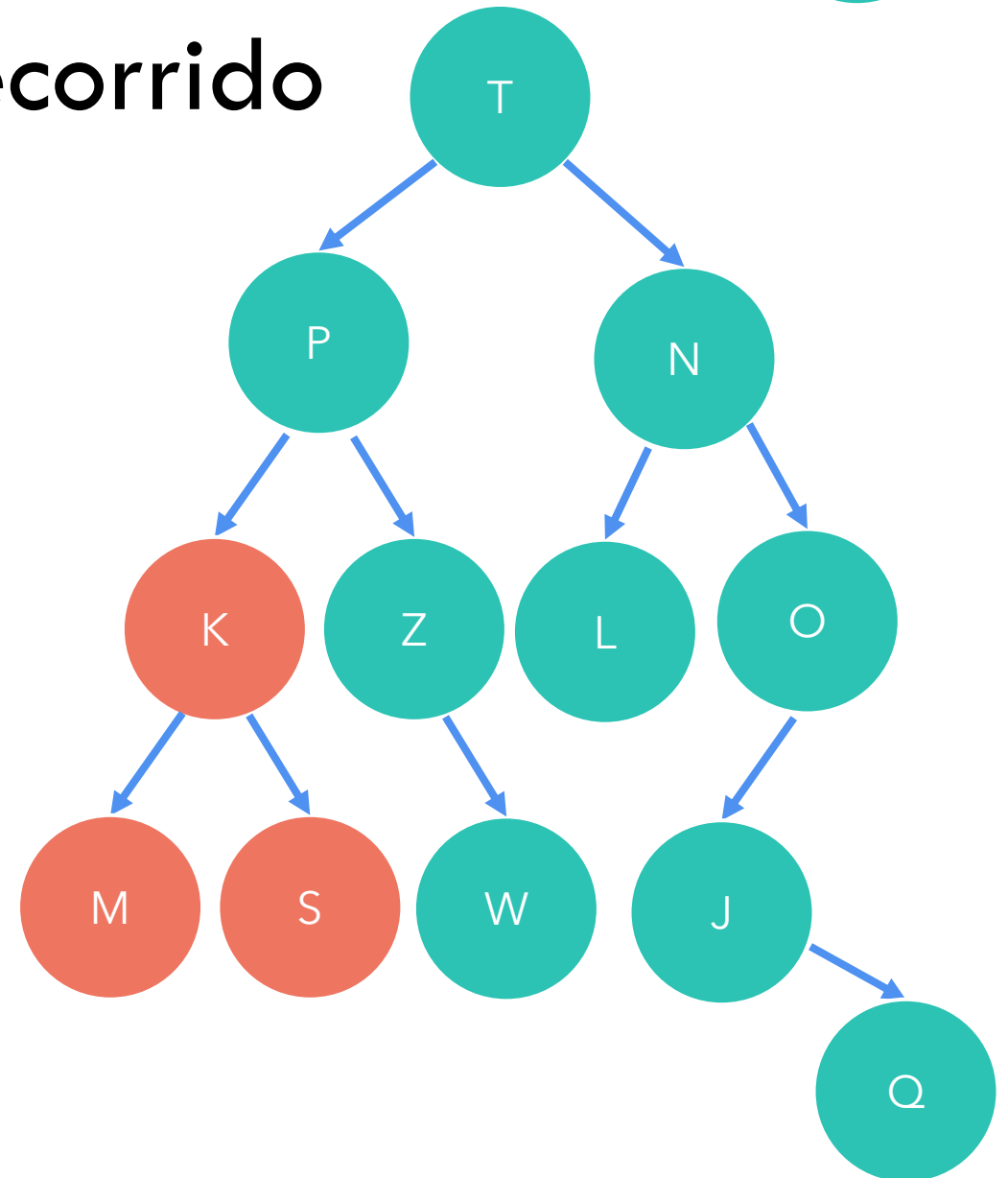
Inorden(T) = M – K –



Arboles – Algoritmos de Recorrido

Algoritmo in-orden:

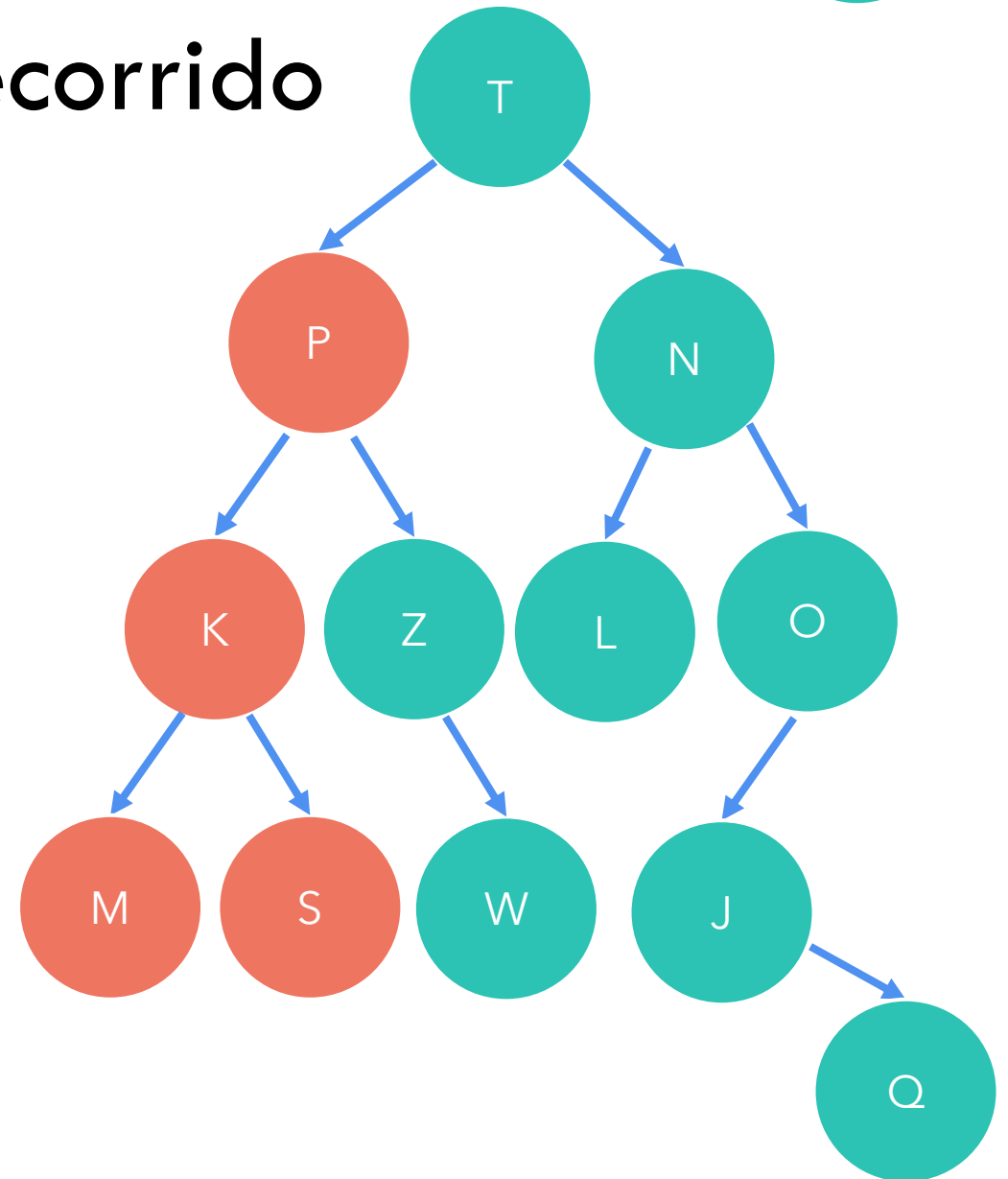
$\text{Inorden}(T) = M - K - S$



Arboles – Algoritmos de Recorrido

Algoritmo in-orden:

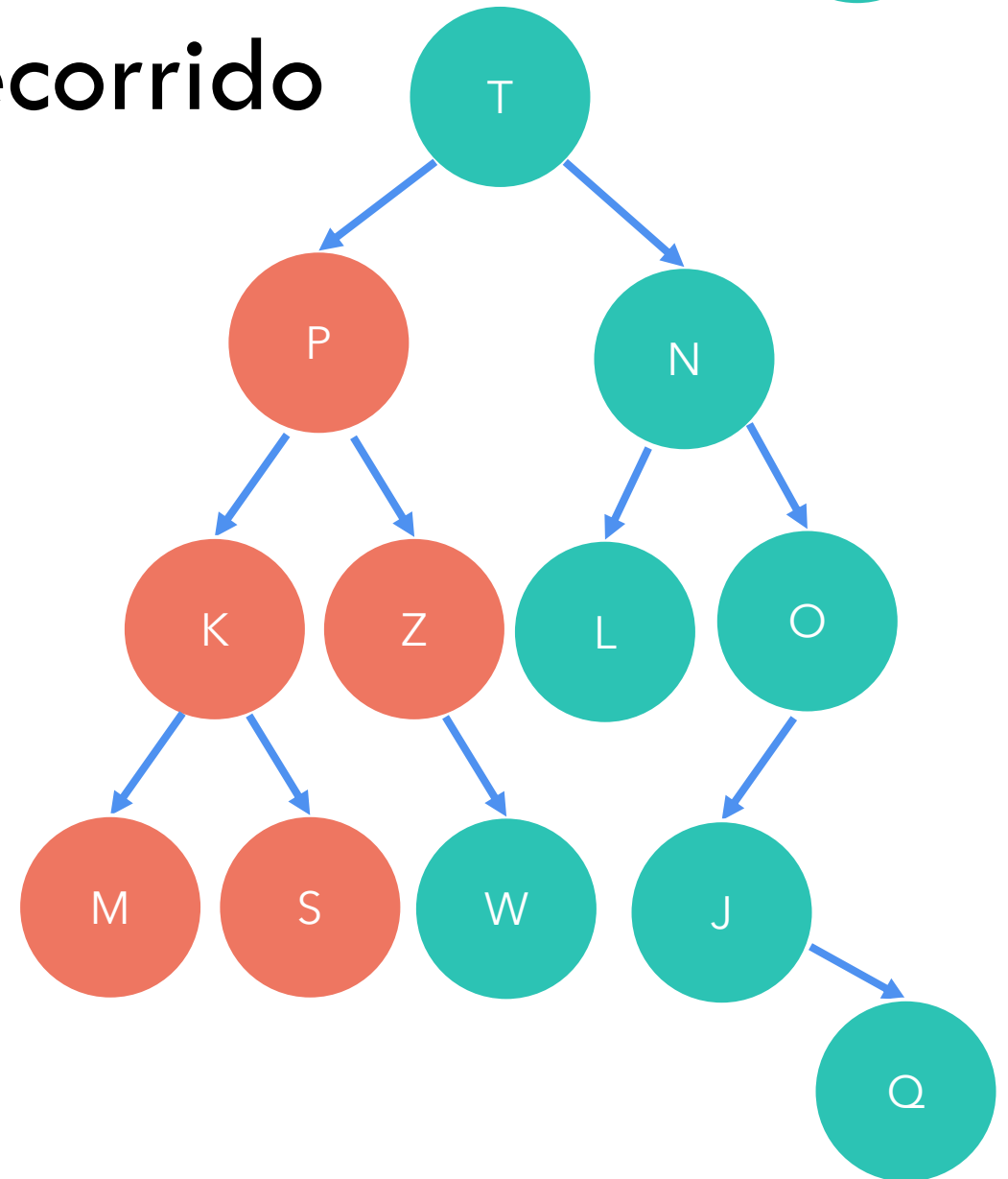
$\text{Inorden}(T) = M - K - S - P$



Arboles – Algoritmos de Recorrido

Algoritmo in-orden:

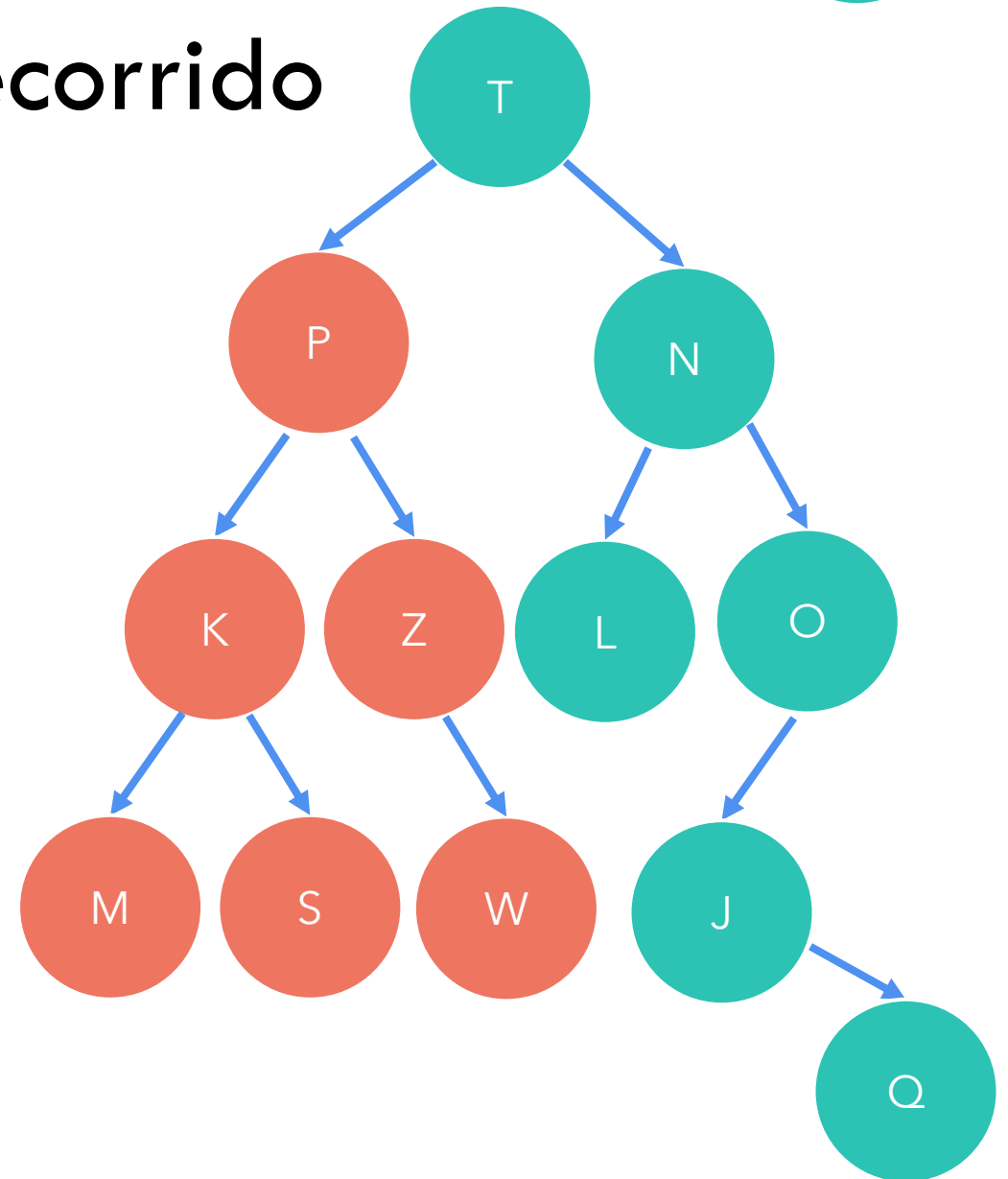
Inorden(T) = M – K – S-P-Z



Arboles – Algoritmos de Recorrido

Algoritmo in-orden:

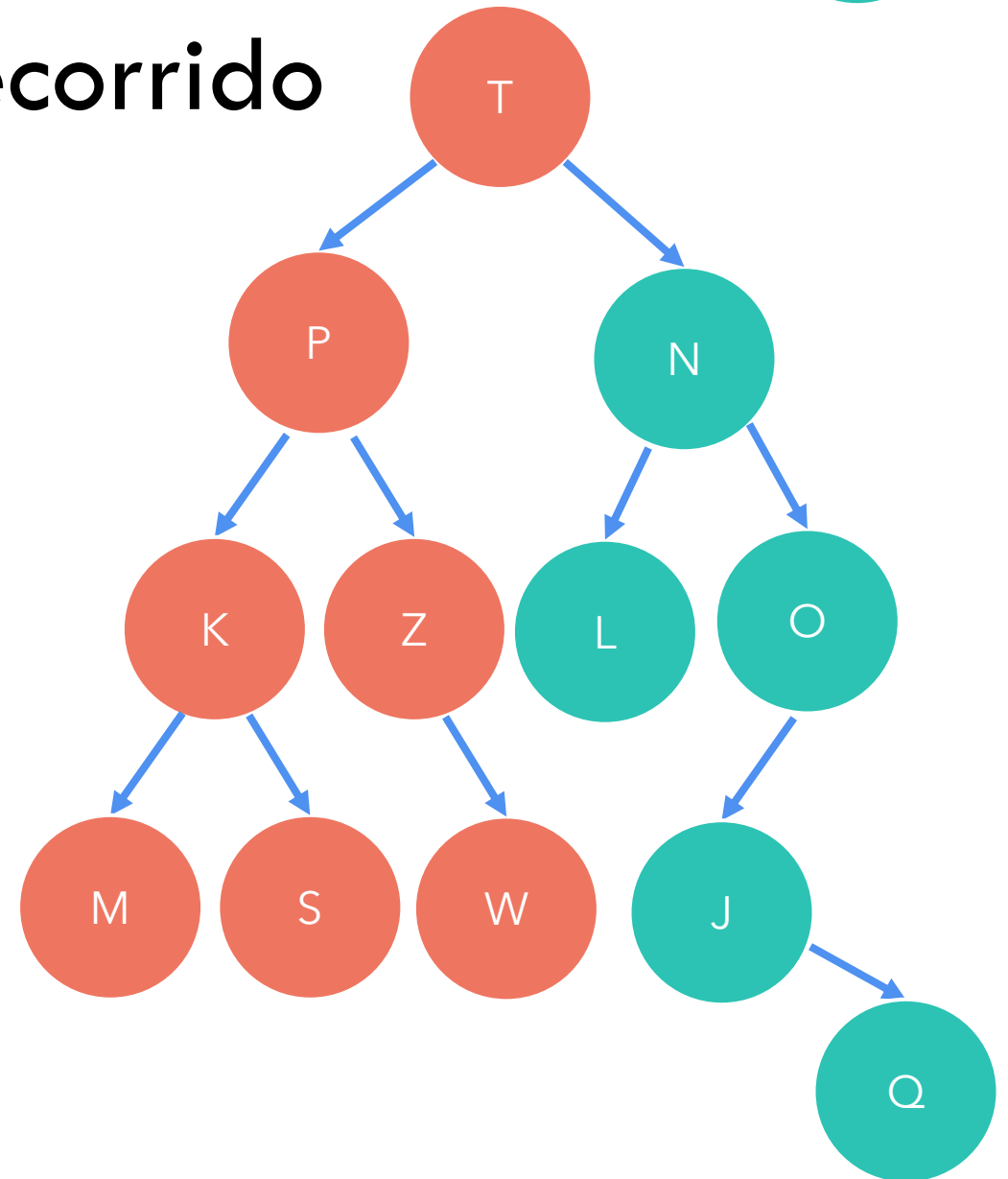
$\text{Inorden}(T) = M - K - S - P - Z - W$



Arboles – Algoritmos de Recorrido

Algoritmo in-orden:

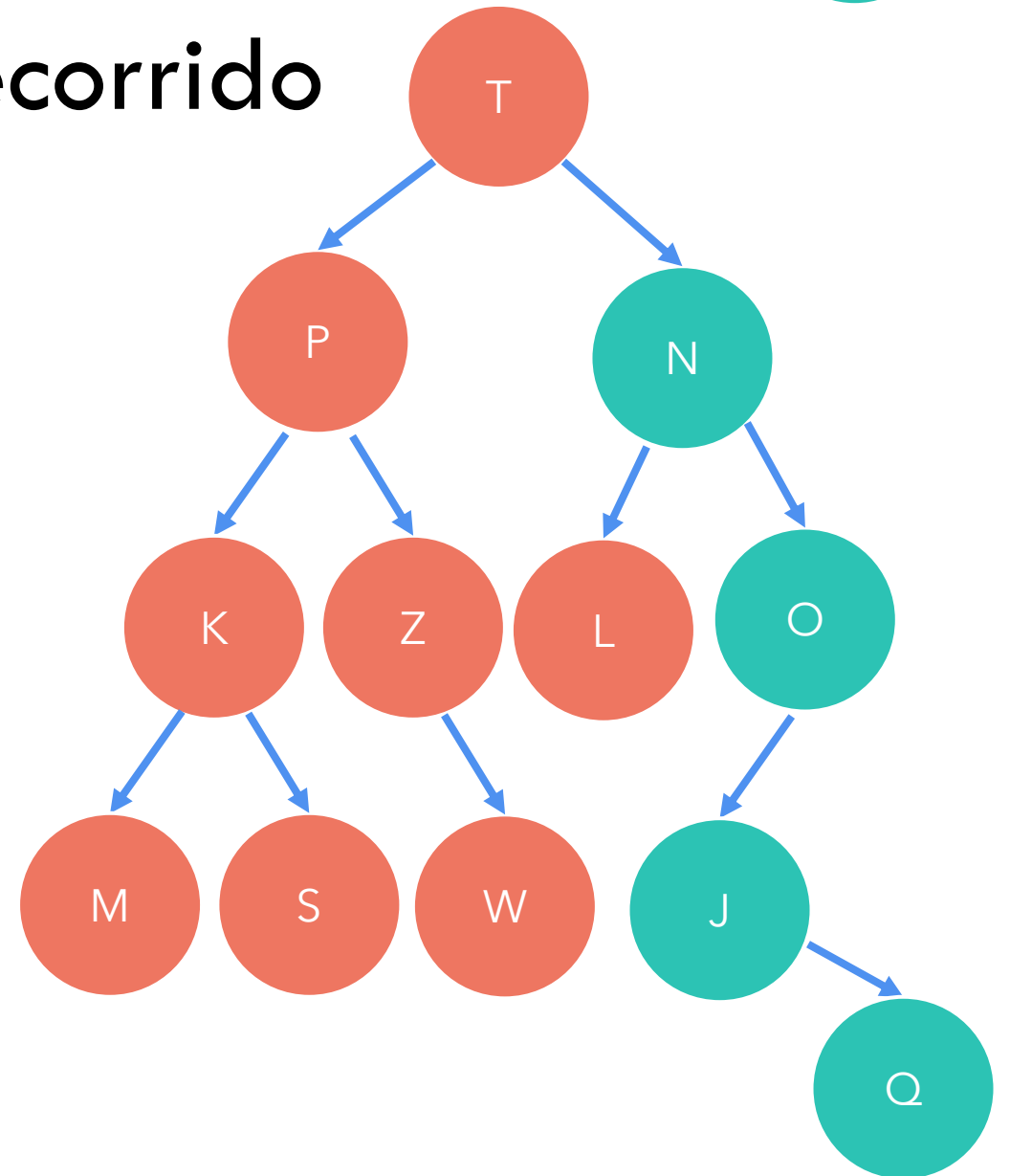
$\text{Inorden}(T) = M - K - S - P - Z - W - T$



Arboles – Algoritmos de Recorrido

Algoritmo in-orden:

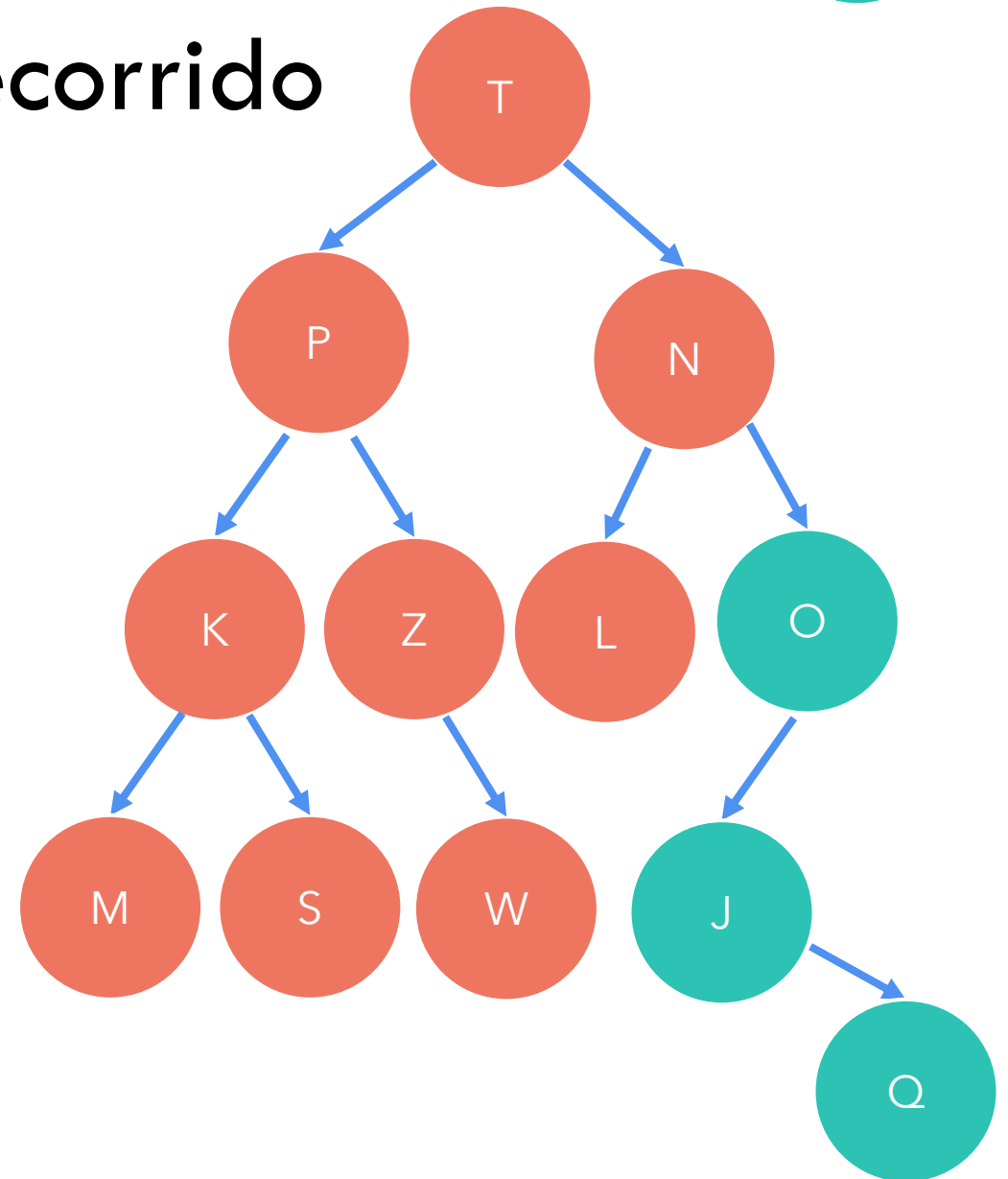
$\text{Inorden}(T) = M - K - S - P - Z - W - T - L$



Arboles – Algoritmos de Recorrido

Algoritmo in-orden:

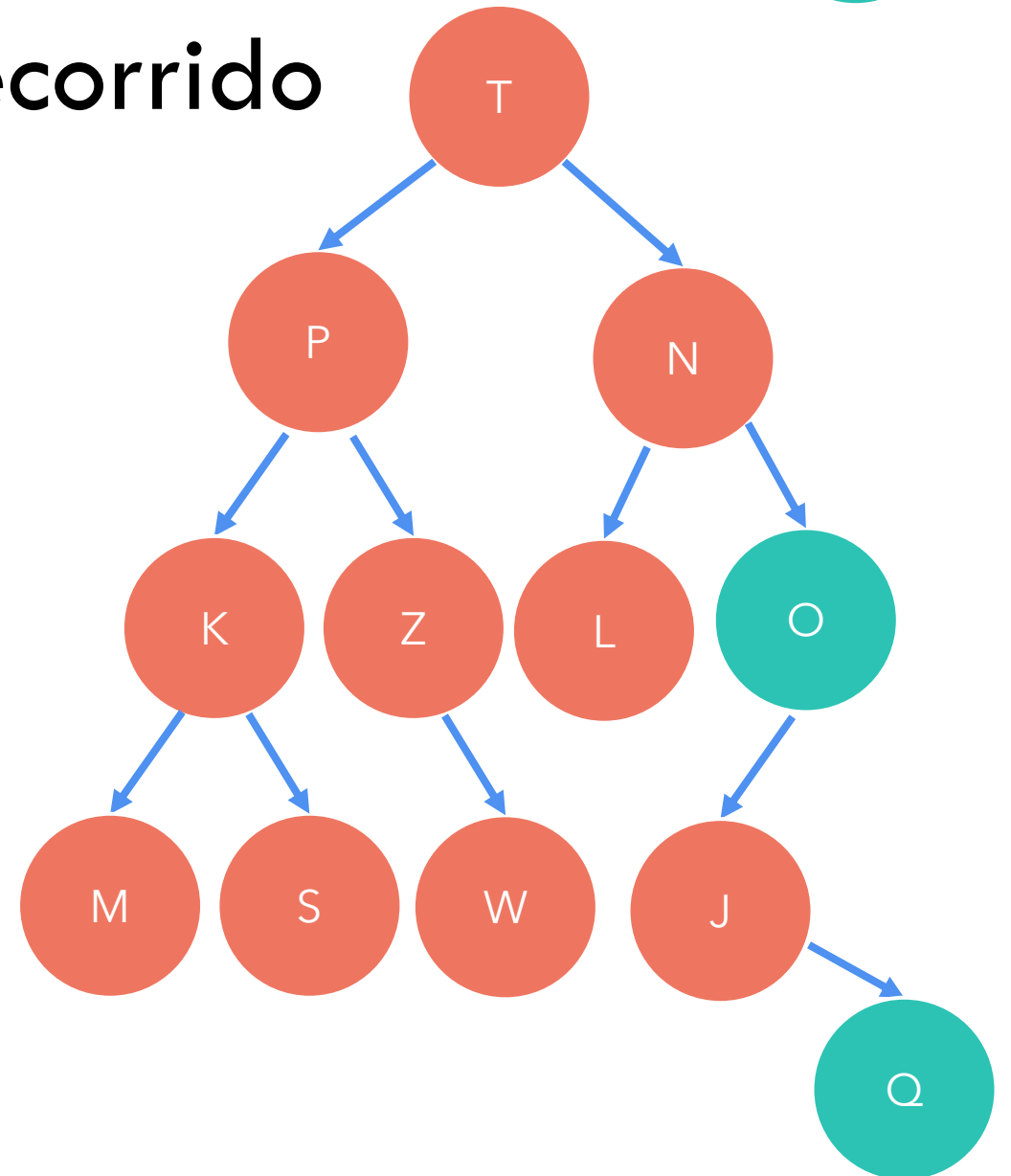
Inorden(T) = M – K – S-P-Z-W-T-L-N



Arboles – Algoritmos de Recorrido

Algoritmo in-orden:

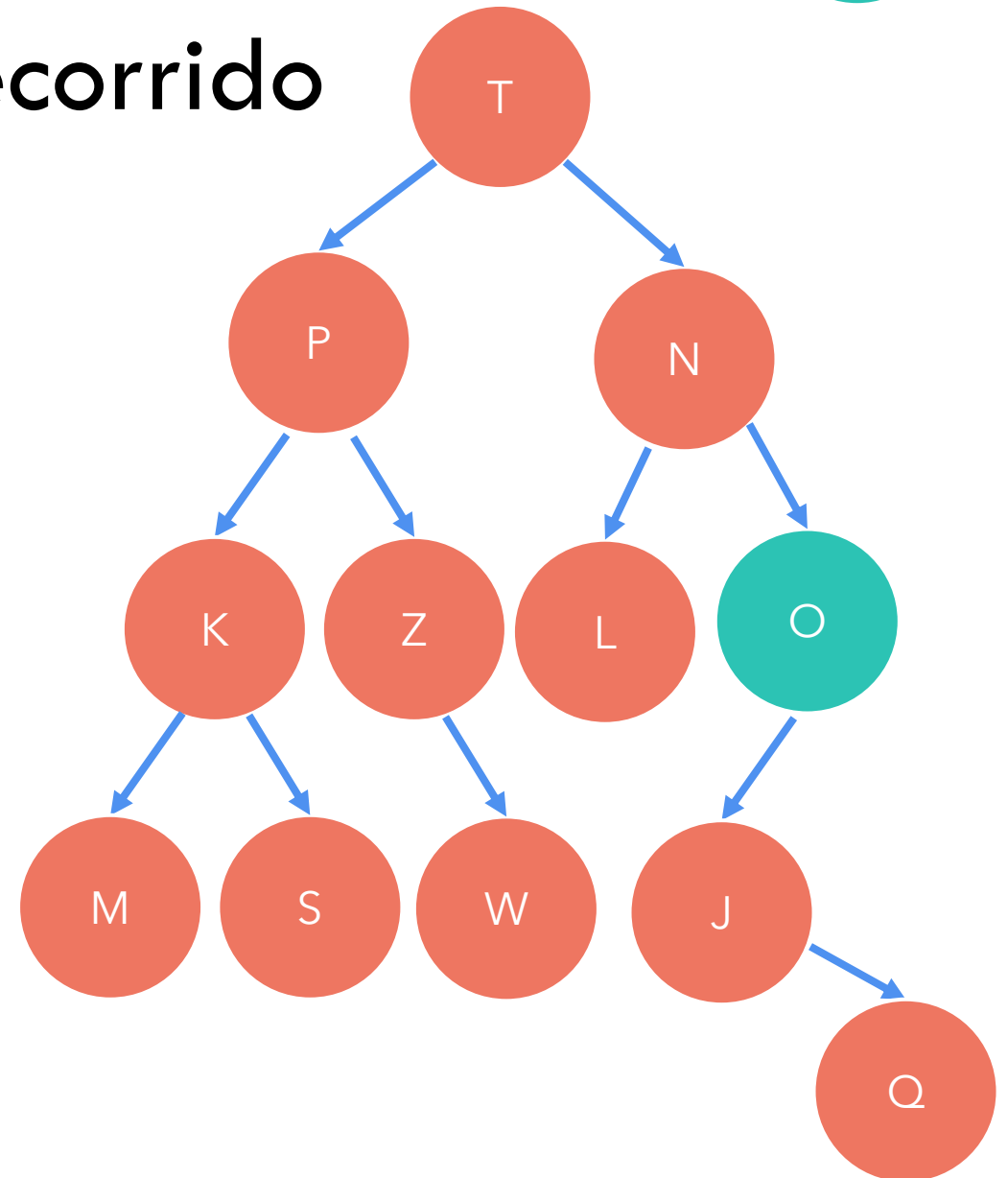
Inorden(T) = M – K – S-P-Z-W-T-L-N-J



Arboles – Algoritmos de Recorrido

Algoritmo in-orden:

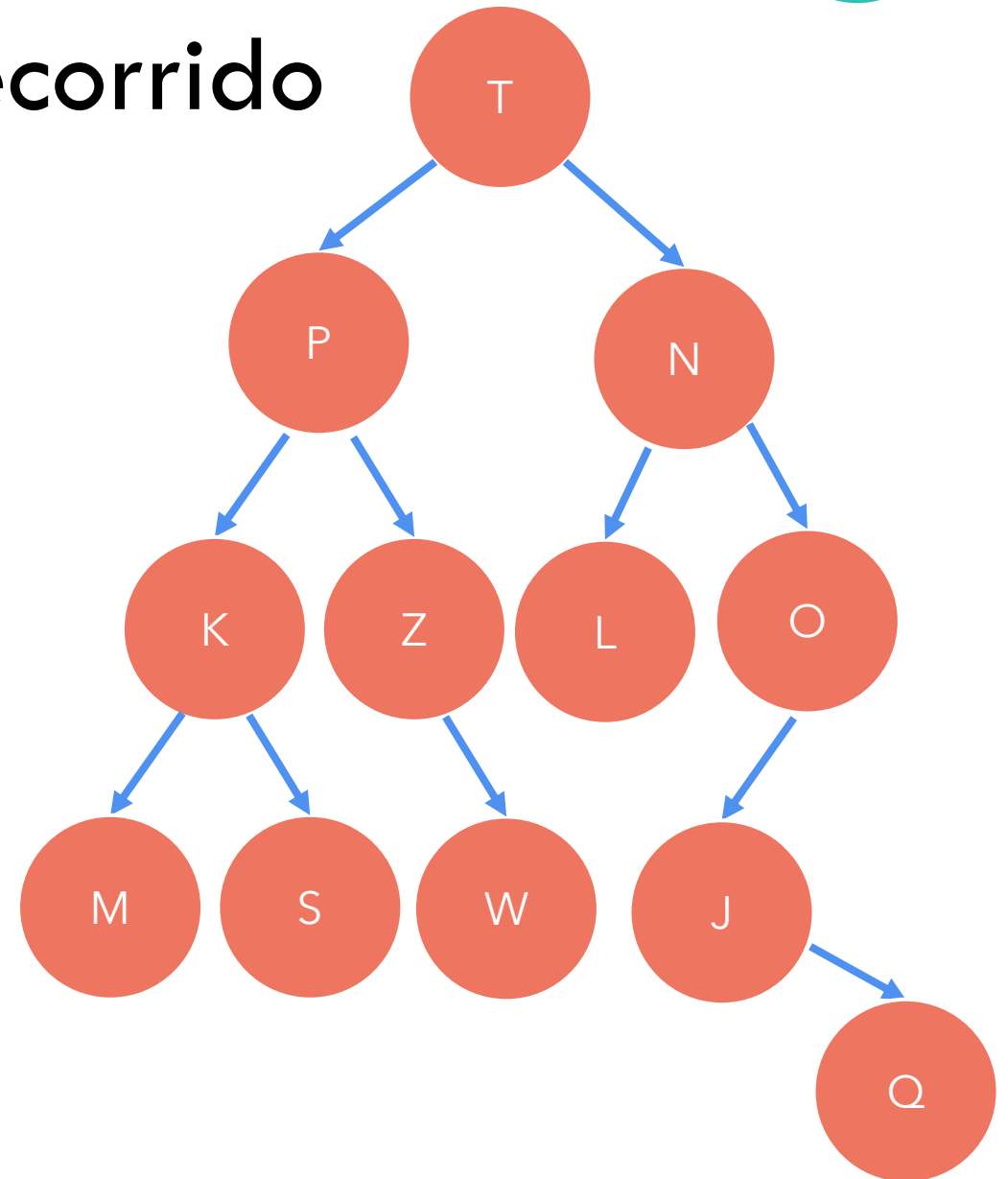
$\text{Inorden}(T) = M - K - S - P - Z - W - T - L - N - J - Q$



Arboles – Algoritmos de Recorrido

Algoritmo in-orden:

Inorden(T) = M – K – S-P-Z-W-T-L-N-J-Q-O



Arboles – Algoritmos de Recorrido

Algoritmo in-orden:

Seudocodigo

Inorder(BinaryTree T, Node v)

1. if T.hasLeft(v)
2. Inorder(T,T.left(v))
3. visit(v)
4. if T.hasRight(v)
5. Inorder(T,T.right(v))

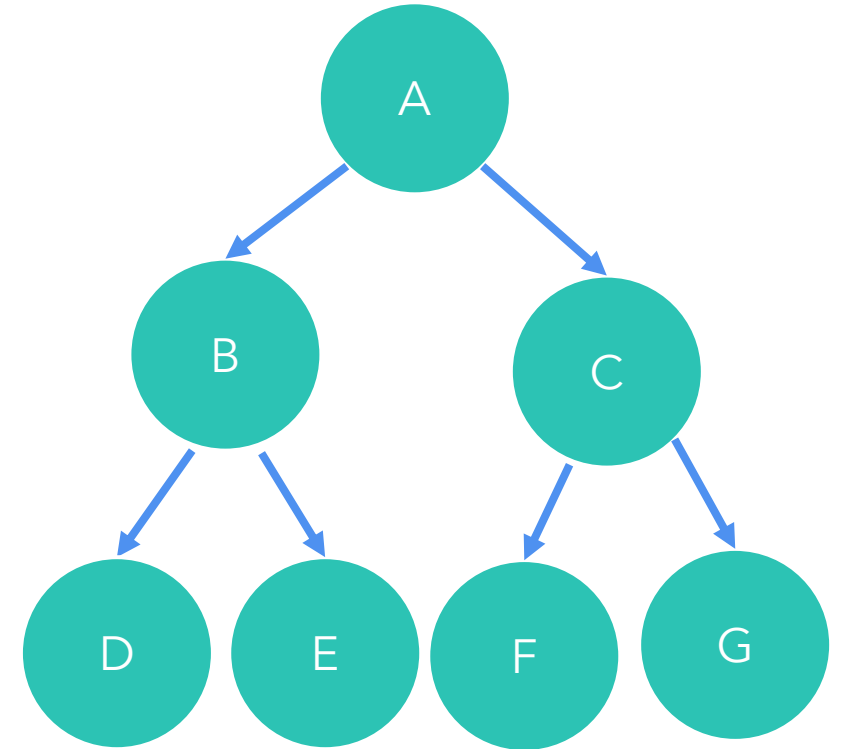
El método visitar se puede modificar de acuerdo con el problema. Ejemplos de operación visitar son:

- Un método para imprimir o guardar en un archivo el contenido del nodo
- Acumular los valores numéricos almacenados en el nodo

Arboles – Algoritmos de Recorrido

Algoritmo pos-orden:

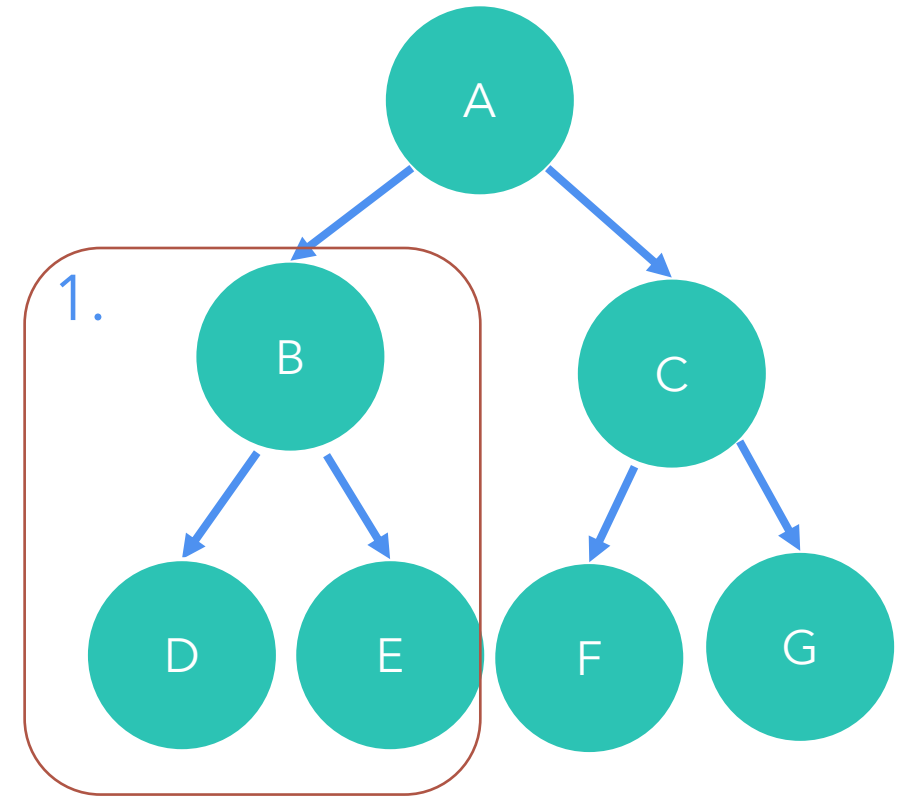
- ❑ En este recorrido primero se visita de forma recursiva el subárbol izquierdo
- ❑ Posteriormente, se visita recursivamente el subárbol derecho
- ❑ Finalmente, se visita la raíz



Arboles – Algoritmos de Recorrido

Algoritmo pos-orden:

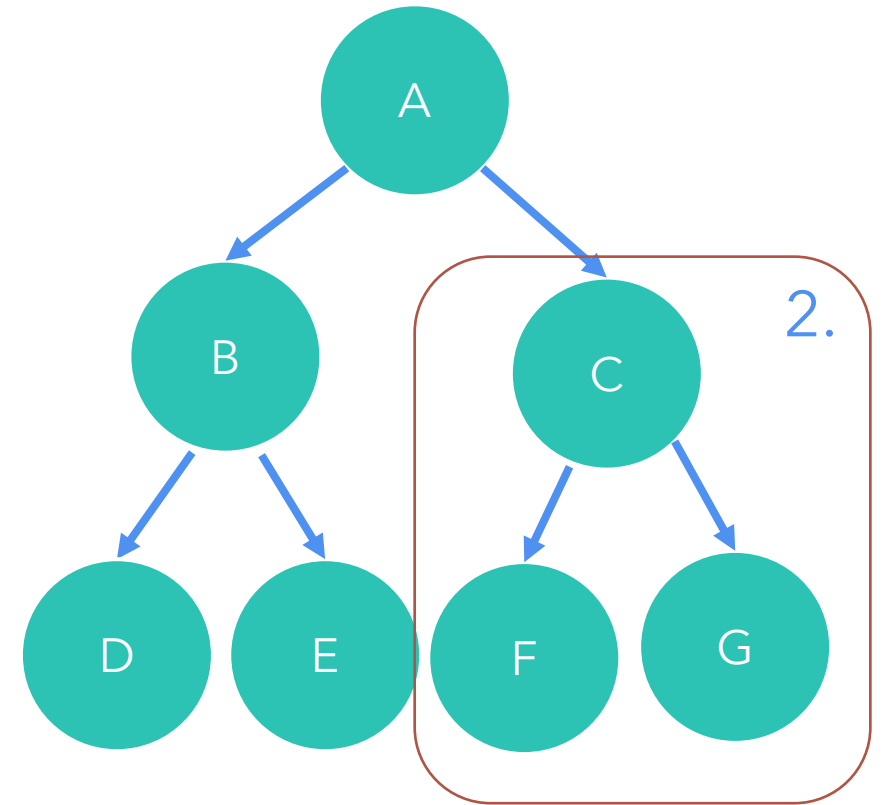
- ❑ En este recorrido primero se visita de forma recursiva el subárbol izquierdo
- ❑ Posteriormente, se visita recursivamente el subárbol derecho
- ❑ Finalmente, se visita la raíz



Arboles – Algoritmos de Recorrido

Algoritmo pos-orden:

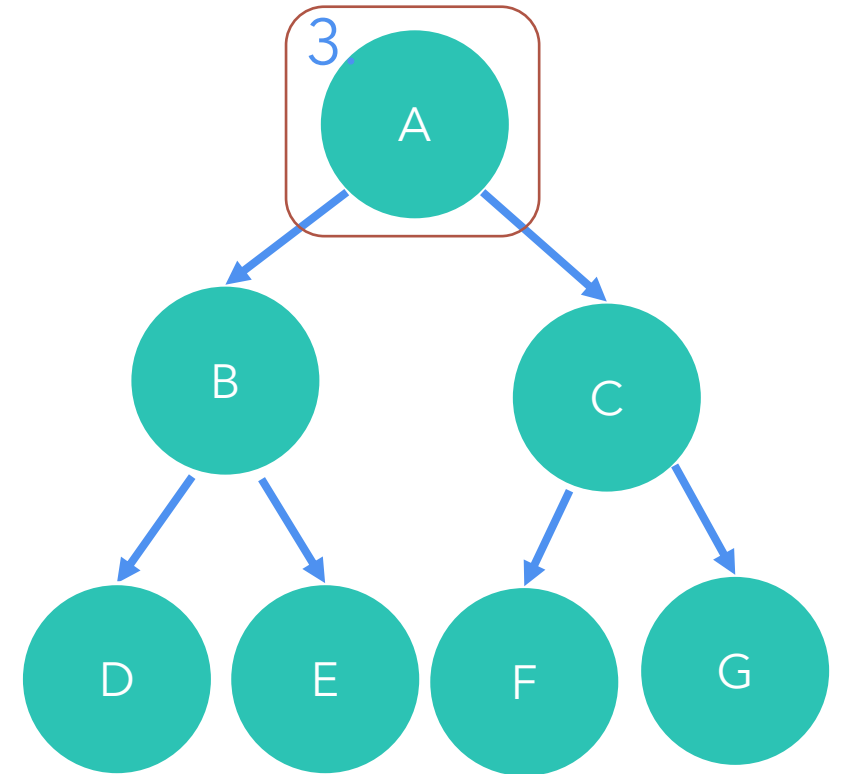
- ❑ En este recorrido primero se visita de forma recursiva el subárbol izquierdo
- ❑ Posteriormente, se visita recursivamente el subárbol derecho
- ❑ Finalmente, se visita la raíz



Arboles – Algoritmos de Recorrido

Algoritmo pos-orden:

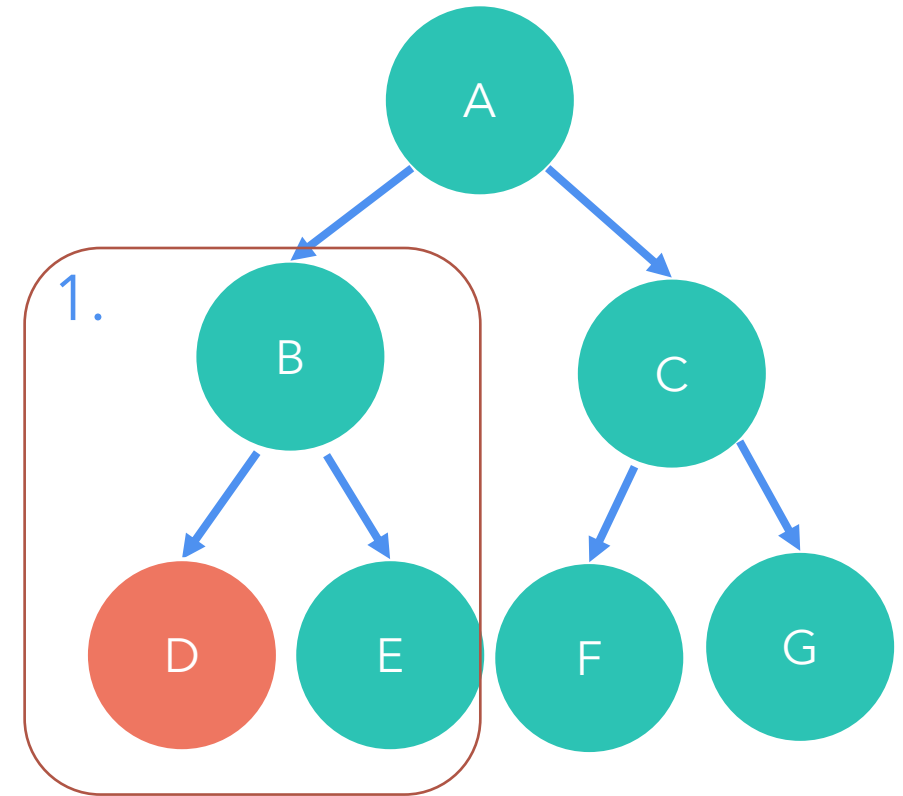
- ❑ En este recorrido primero se visita de forma recursiva el subárbol izquierdo
- ❑ Posteriormente, se visita recursivamente el subárbol derecho
- ❑ Finalmente, se visita la raíz



Arboles – Algoritmos de Recorrido

Algoritmo pos-orden:

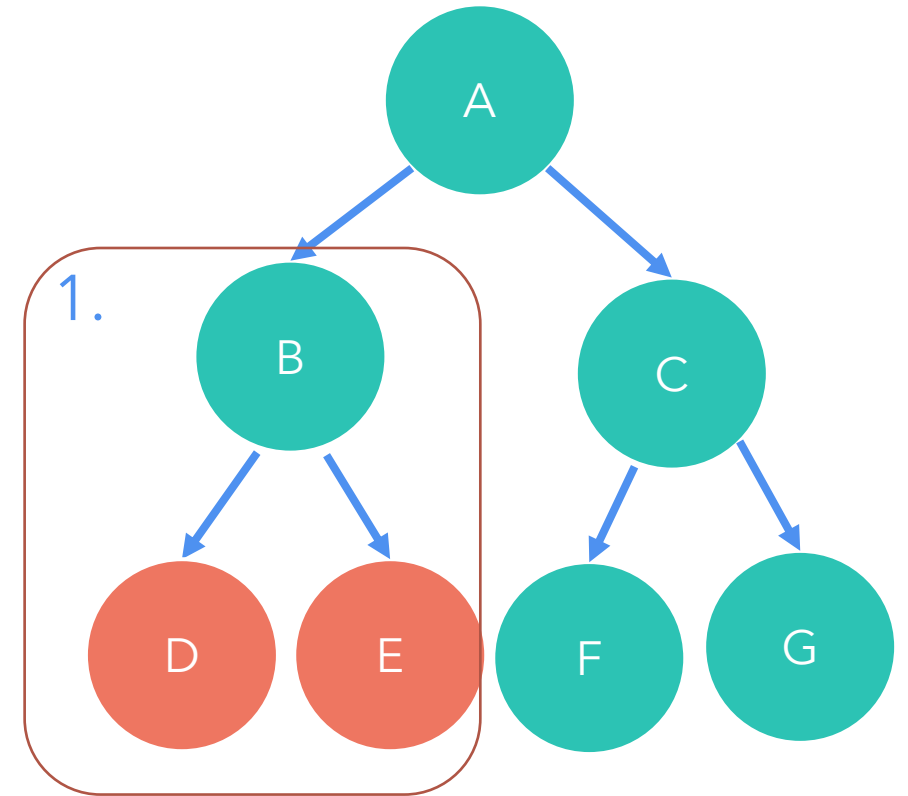
Posorden: D-



Arboles – Algoritmos de Recorrido

Algoritmo pos-orden:

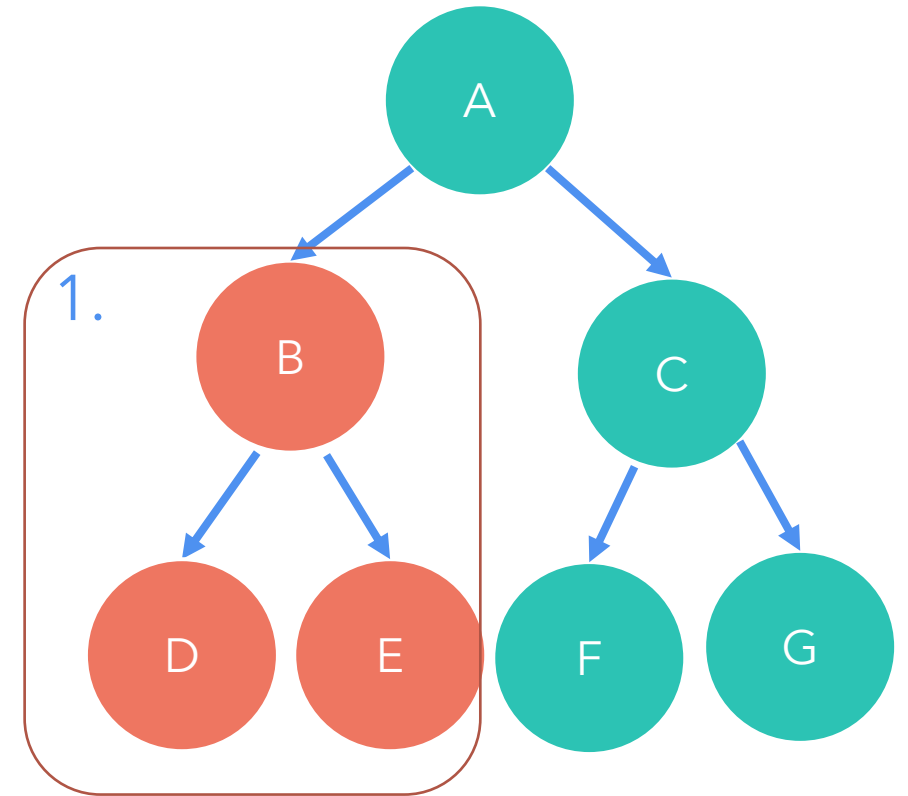
Posorden: D-E



Arboles – Algoritmos de Recorrido

Algoritmo pos-orden:

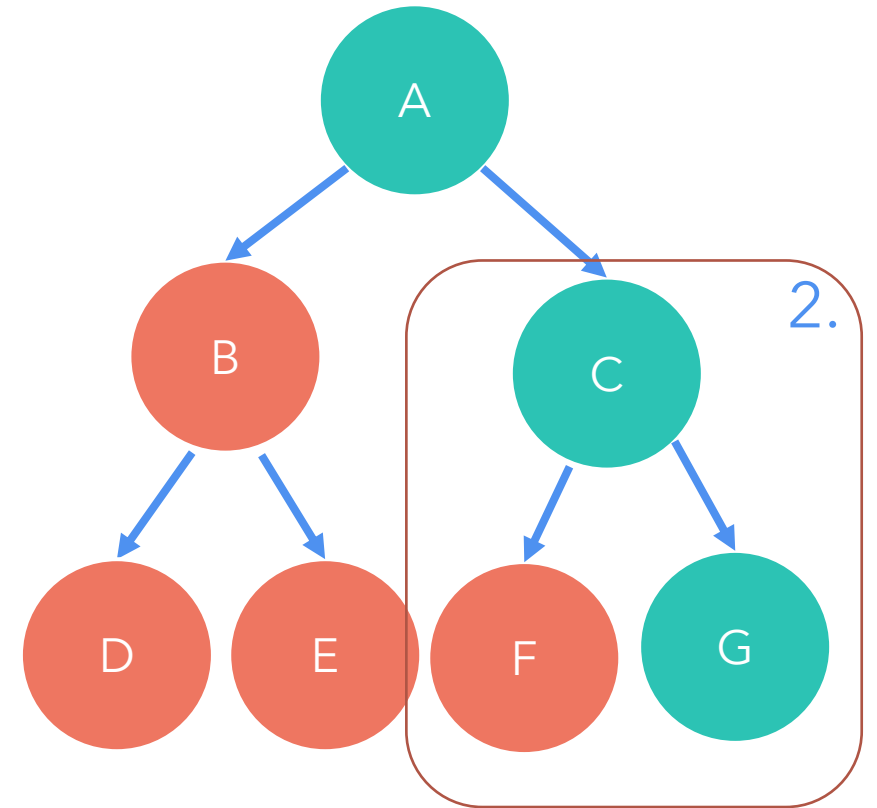
Posorden: D-E-B



Arboles – Algoritmos de Recorrido

Algoritmo pos-orden:

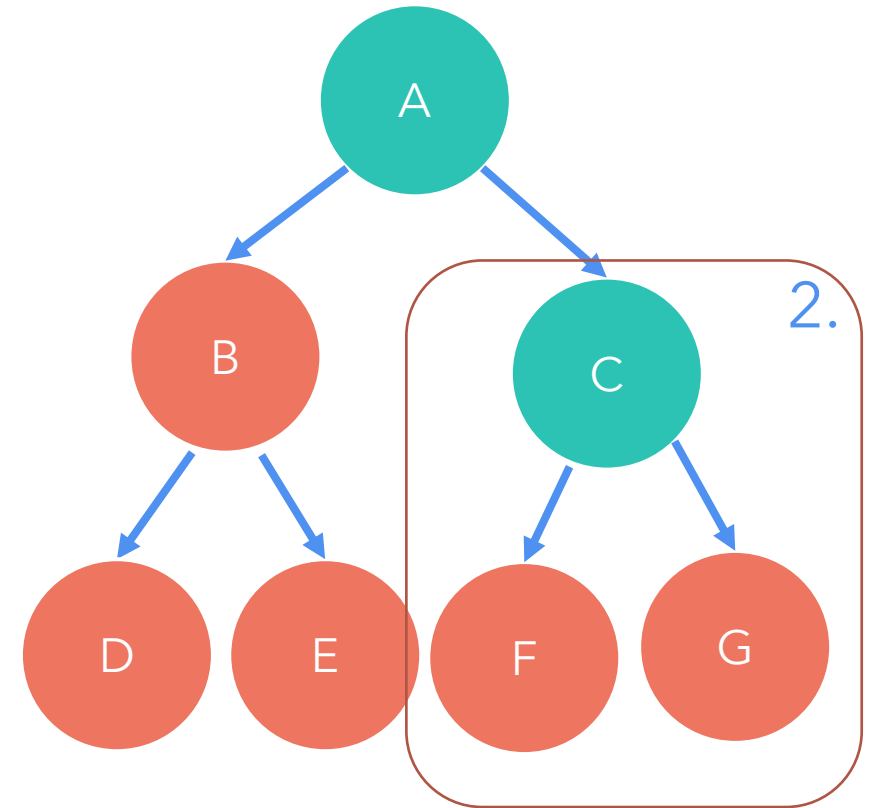
Posorden: D-E-B-F



Arboles – Algoritmos de Recorrido

Algoritmo pos-orden:

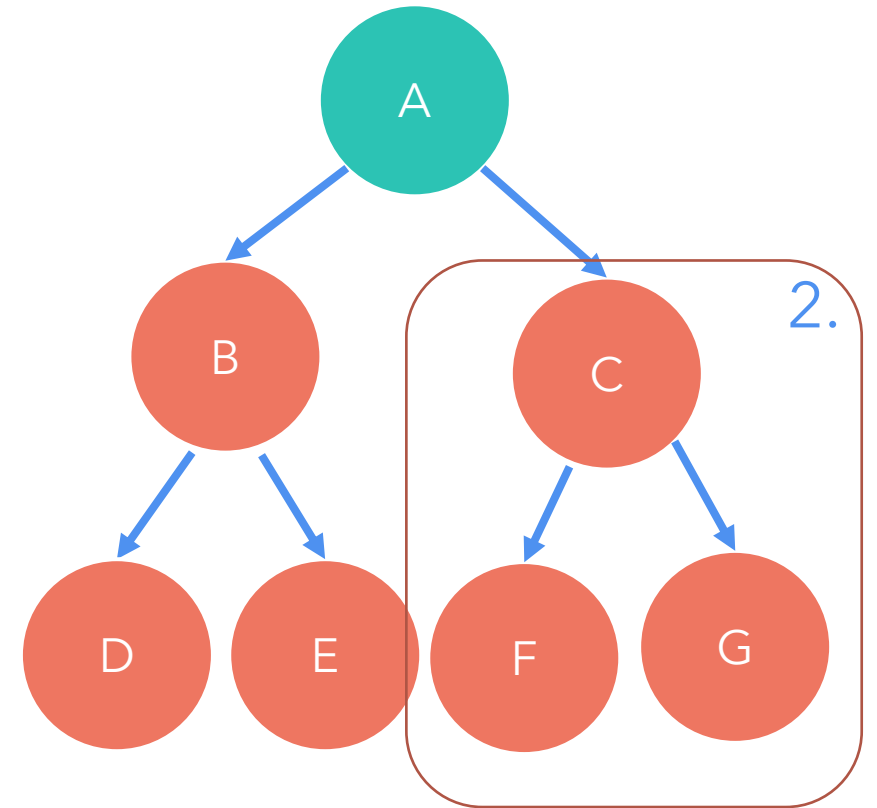
Posorden: D-E-B-F-G



Arboles – Algoritmos de Recorrido

Algoritmo pos-orden:

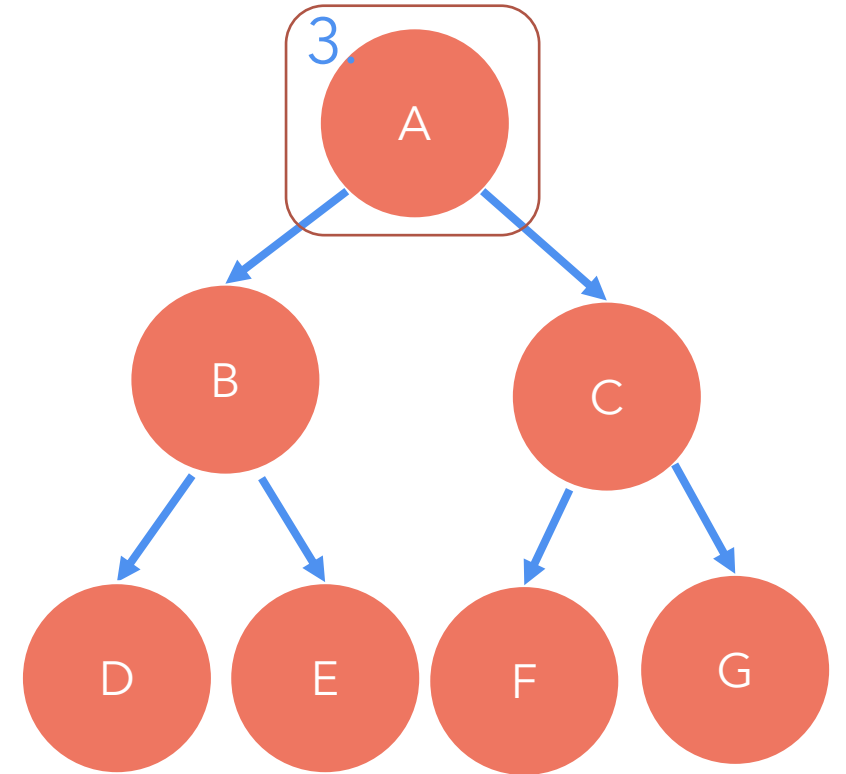
Posorden: D-E-B-F-G-C



Arboles – Algoritmos de Recorrido

Algoritmo pos-orden:

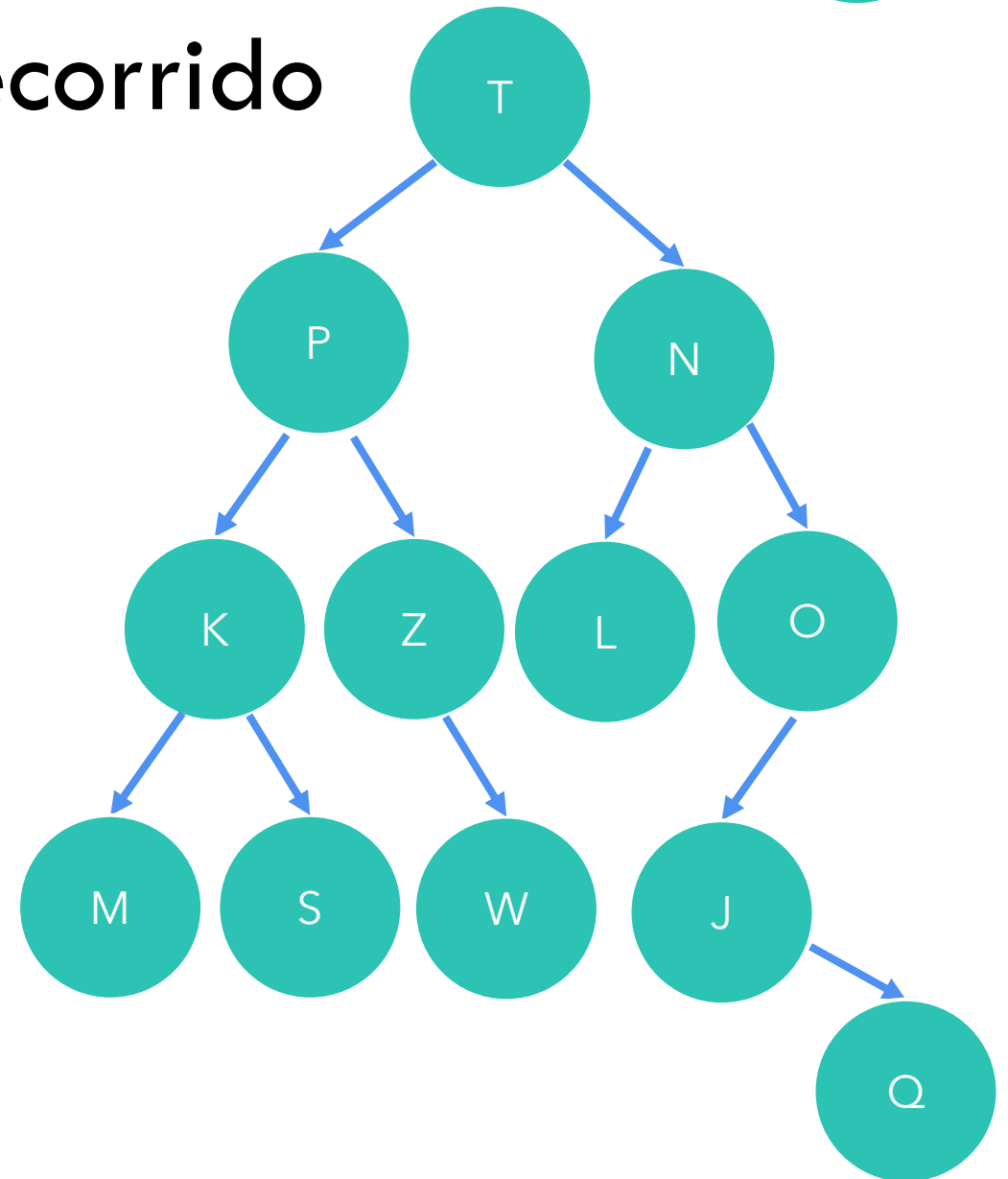
Posorden: D-E-B-F-G-C-A



Arboles – Algoritmos de Recorrido

Algoritmo pos-orden:

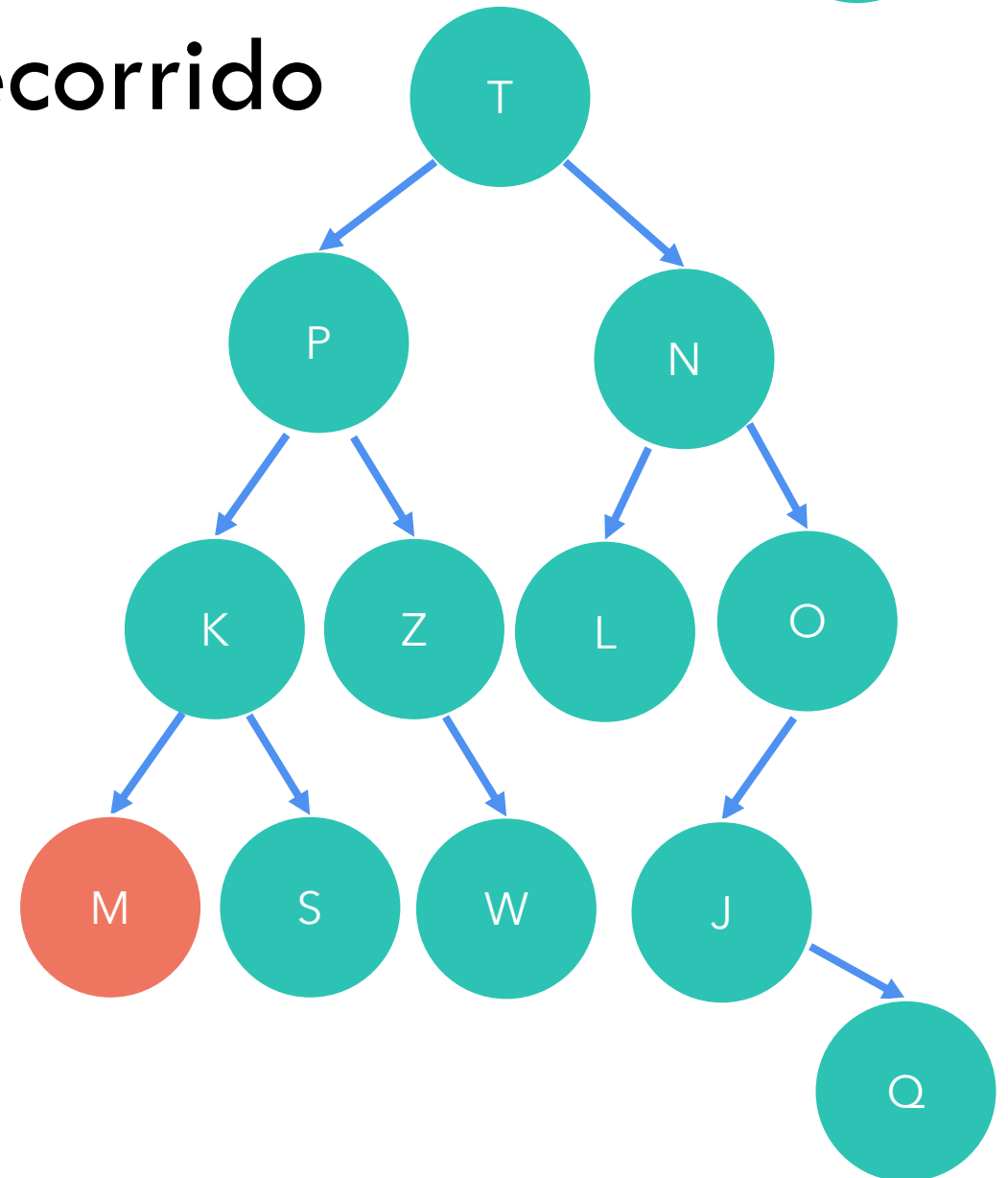
posorden(T) =



Arboles – Algoritmos de Recorrido

Algoritmo pos-orden:

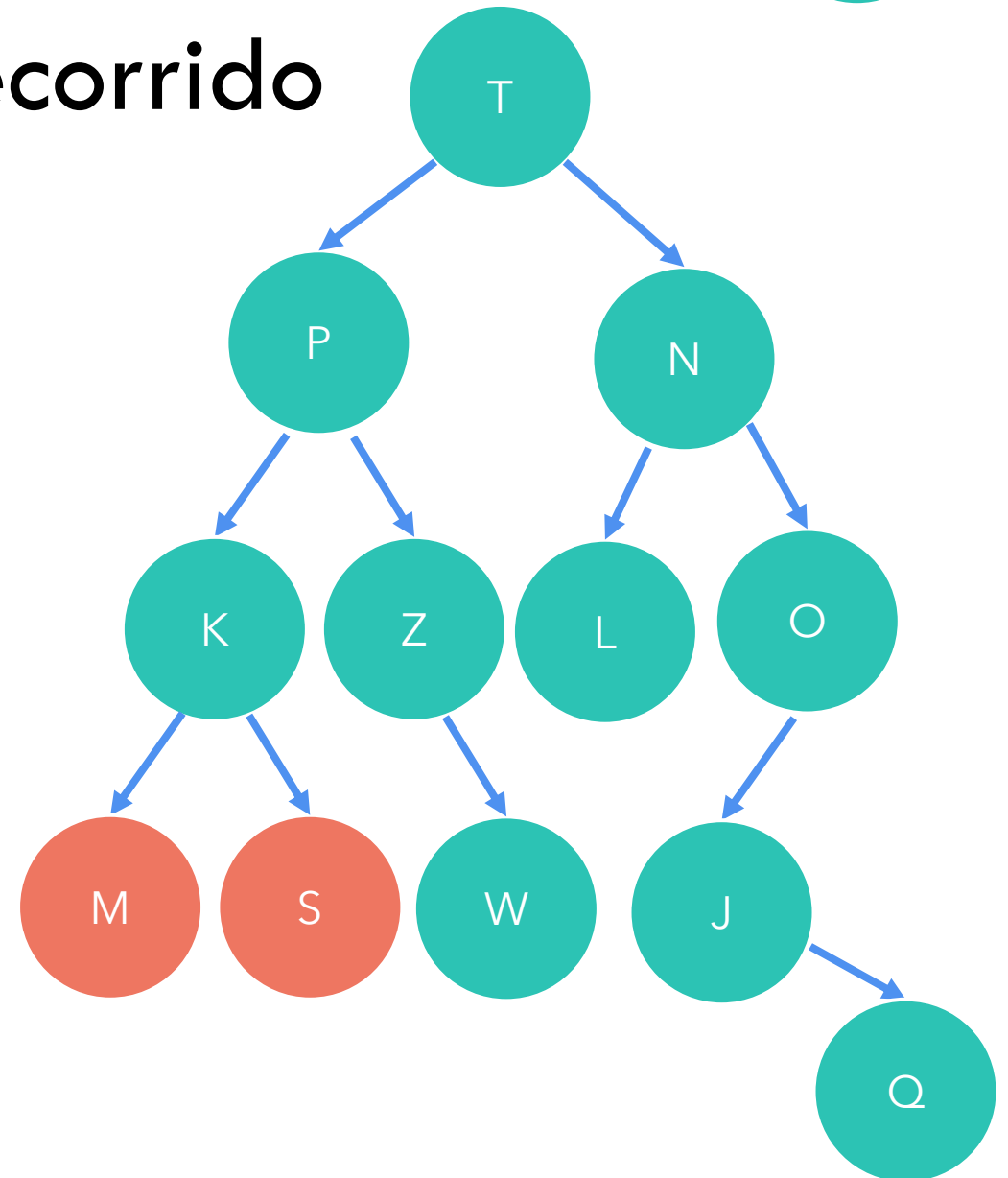
$\text{posorden}(T) = M$



Arboles – Algoritmos de Recorrido

Algoritmo pos-orden:

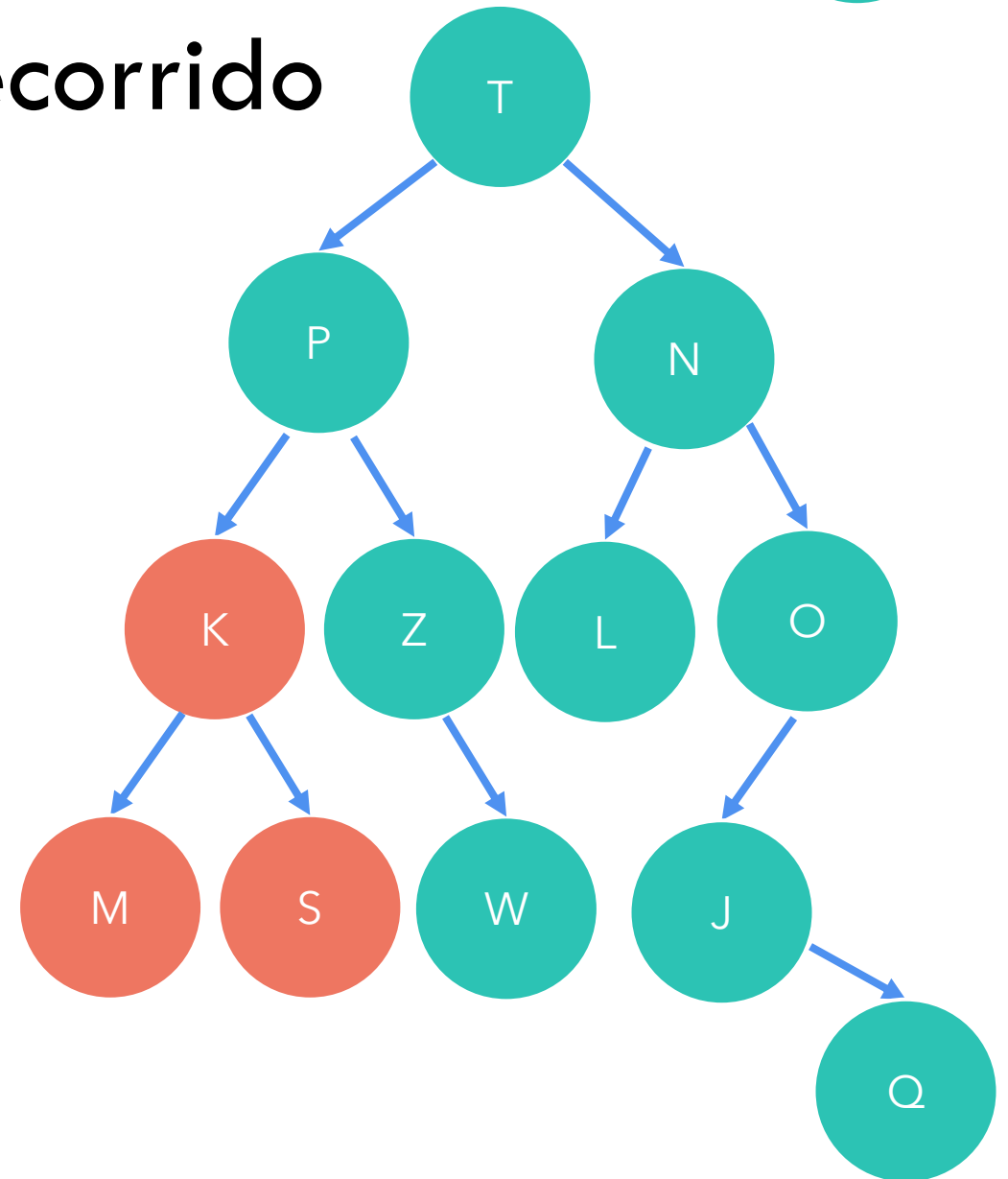
$\text{posorden}(T) = M-S$



Arboles – Algoritmos de Recorrido

Algoritmo pos-orden:

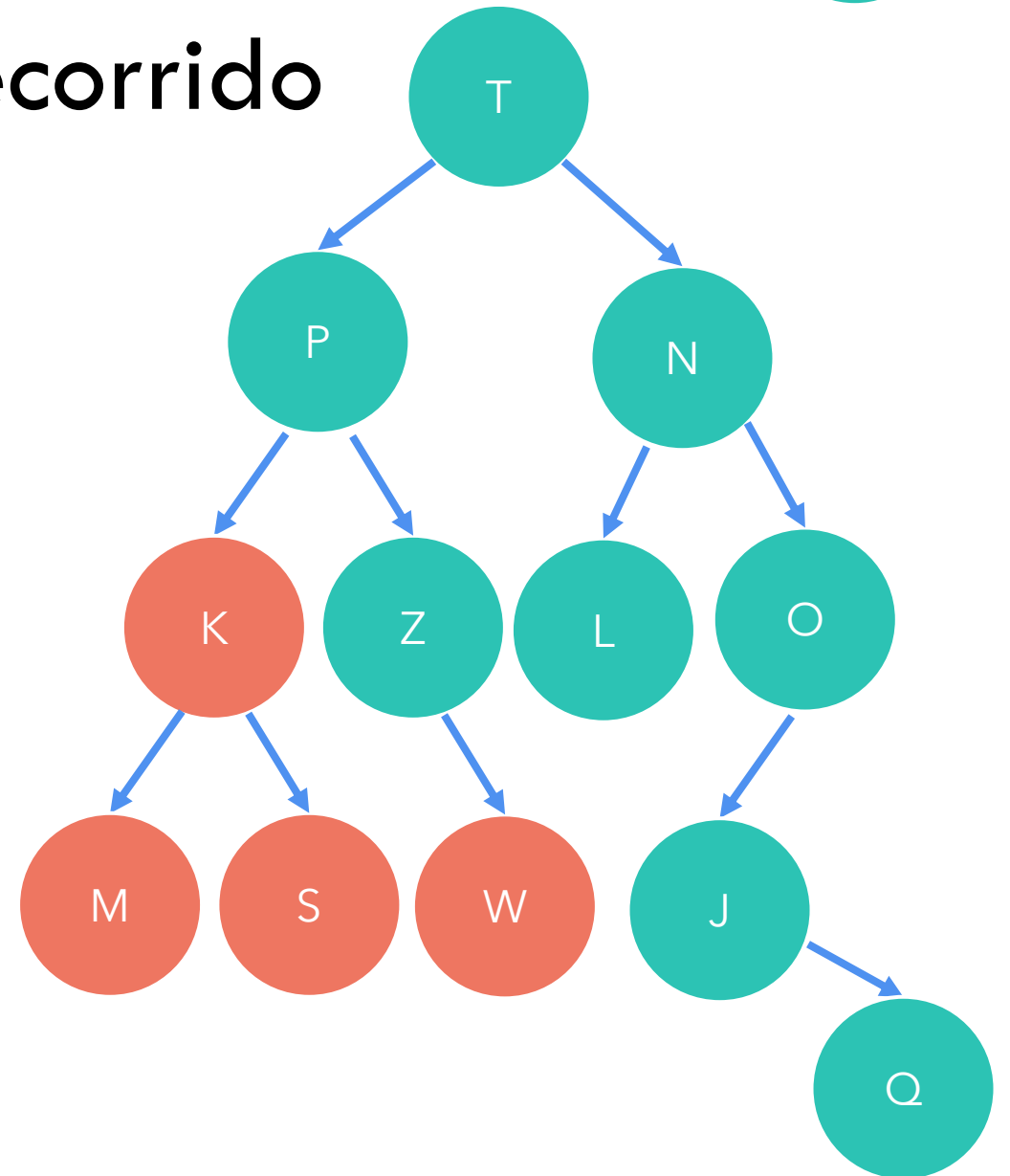
$\text{posorden}(T) = M-S-K$



Arboles – Algoritmos de Recorrido

Algoritmo pos-orden:

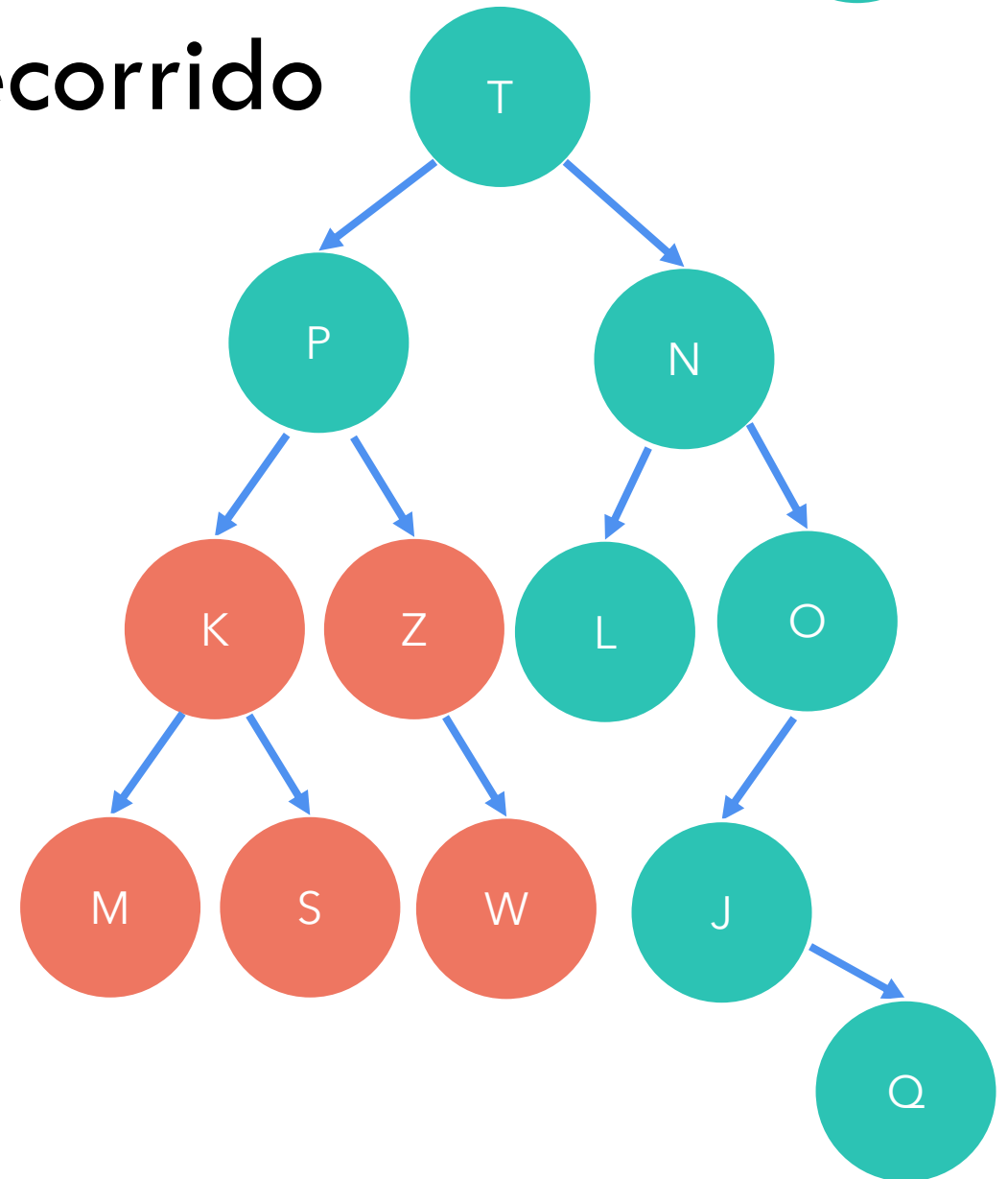
$\text{posorden}(T) = M-S-K-W$



Arboles – Algoritmos de Recorrido

Algoritmo pos-orden:

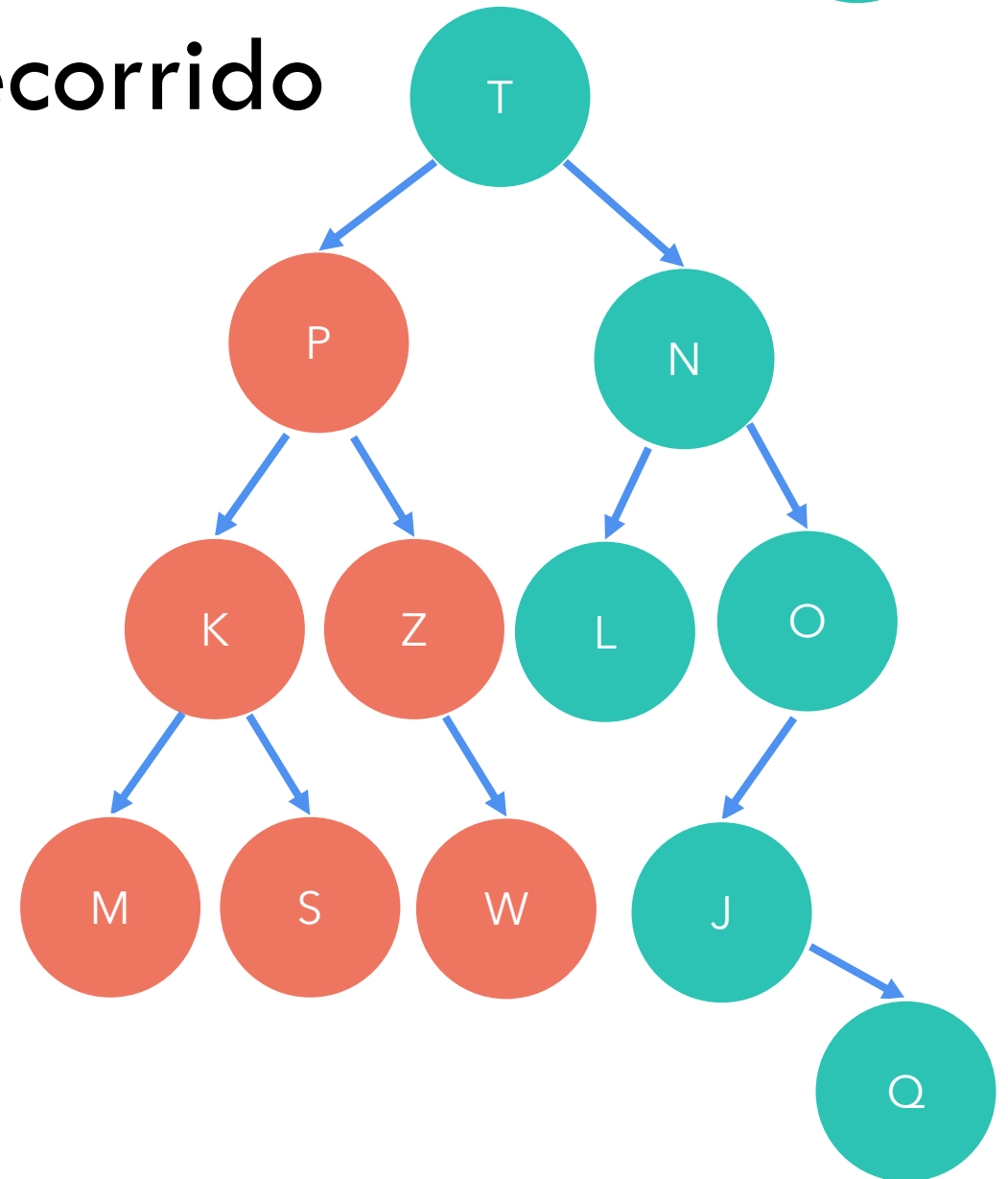
$\text{posorden}(T) = M-S-K-W-Z$



Arboles – Algoritmos de Recorrido

Algoritmo pos-orden:

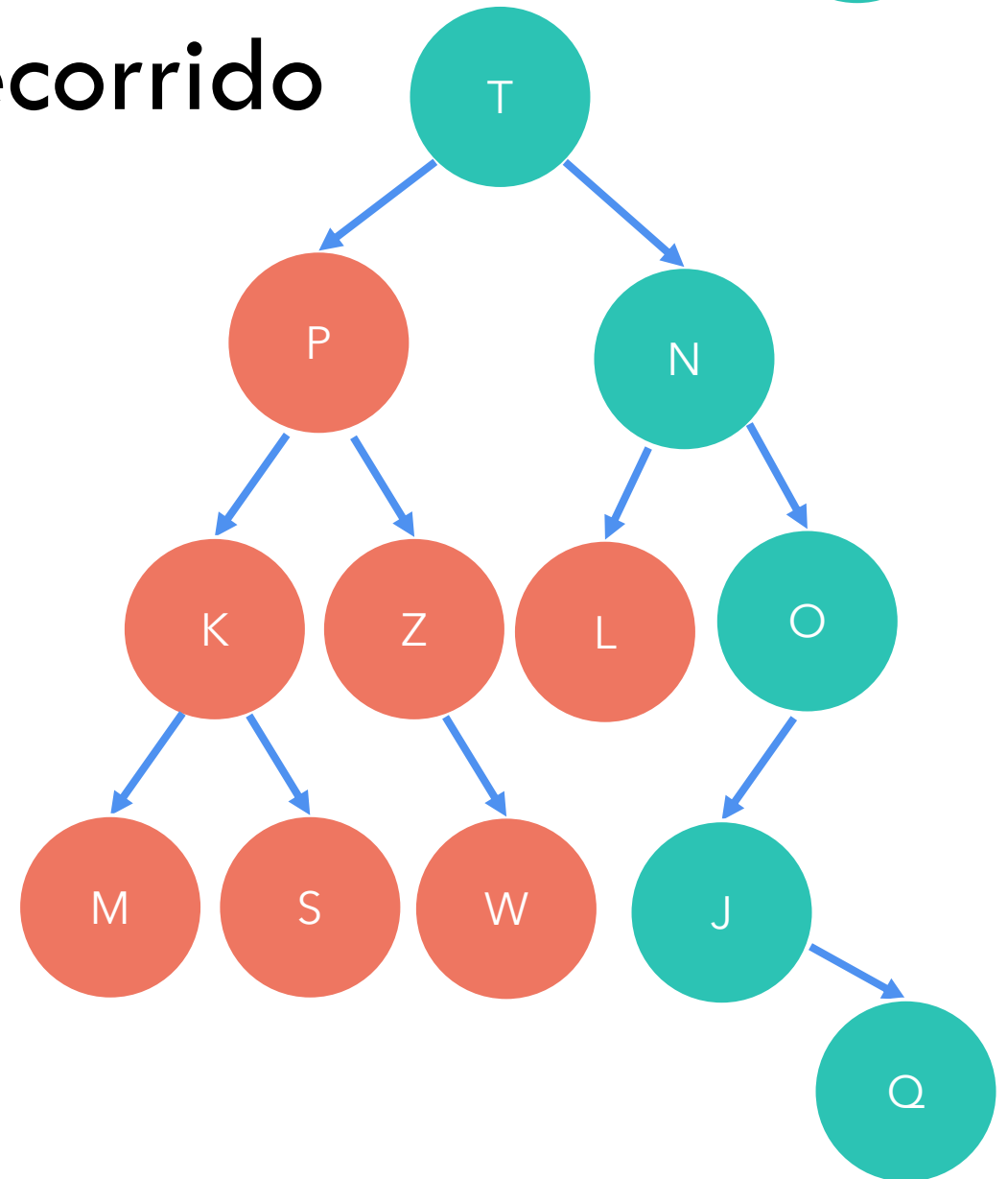
$\text{posorden}(T) = M-S-K-W-Z-P$



Arboles – Algoritmos de Recorrido

Algoritmo pos-orden:

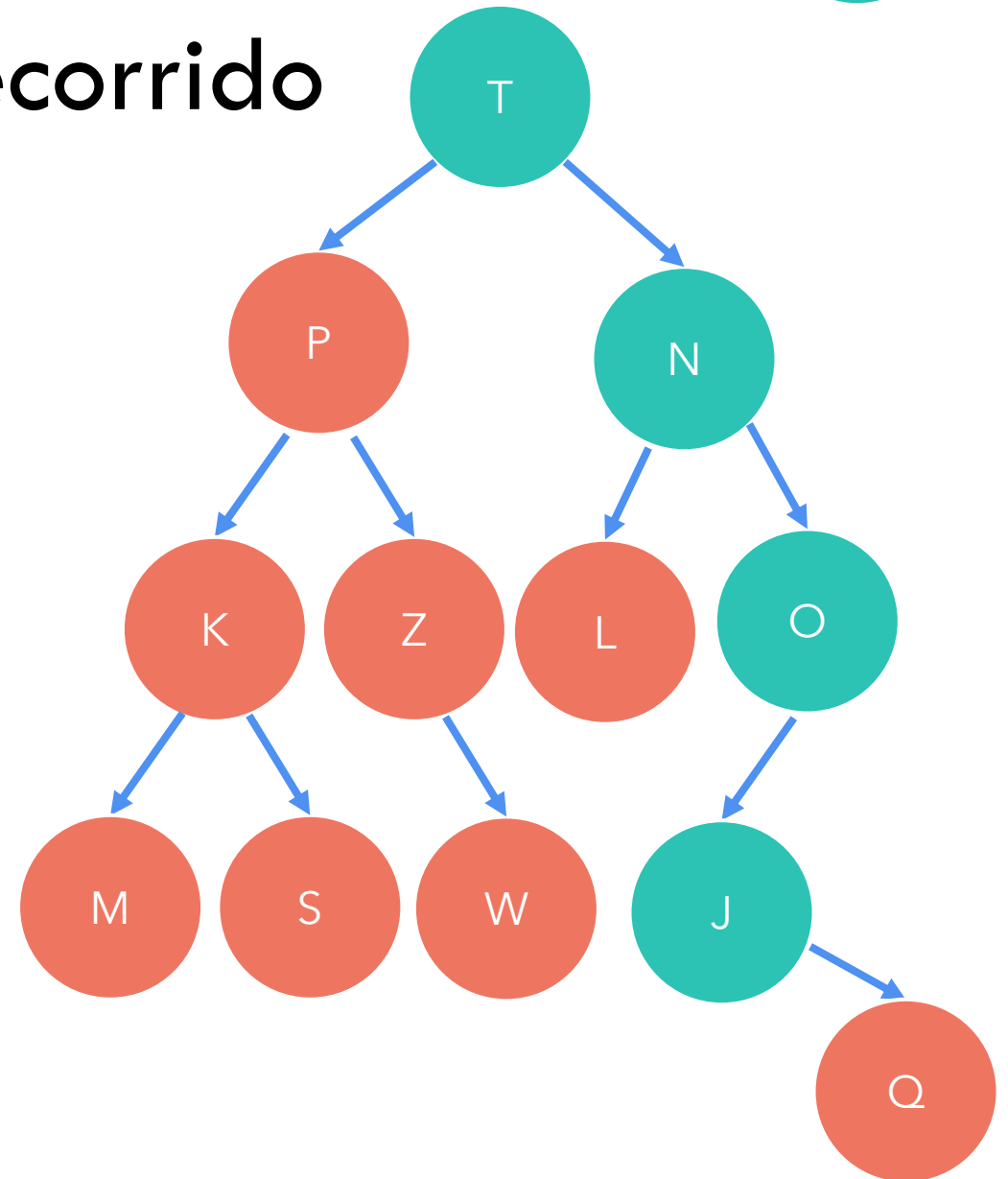
$\text{posorden}(T) = M-S-K-W-Z-P-L$



Arboles – Algoritmos de Recorrido

Algoritmo pos-orden:

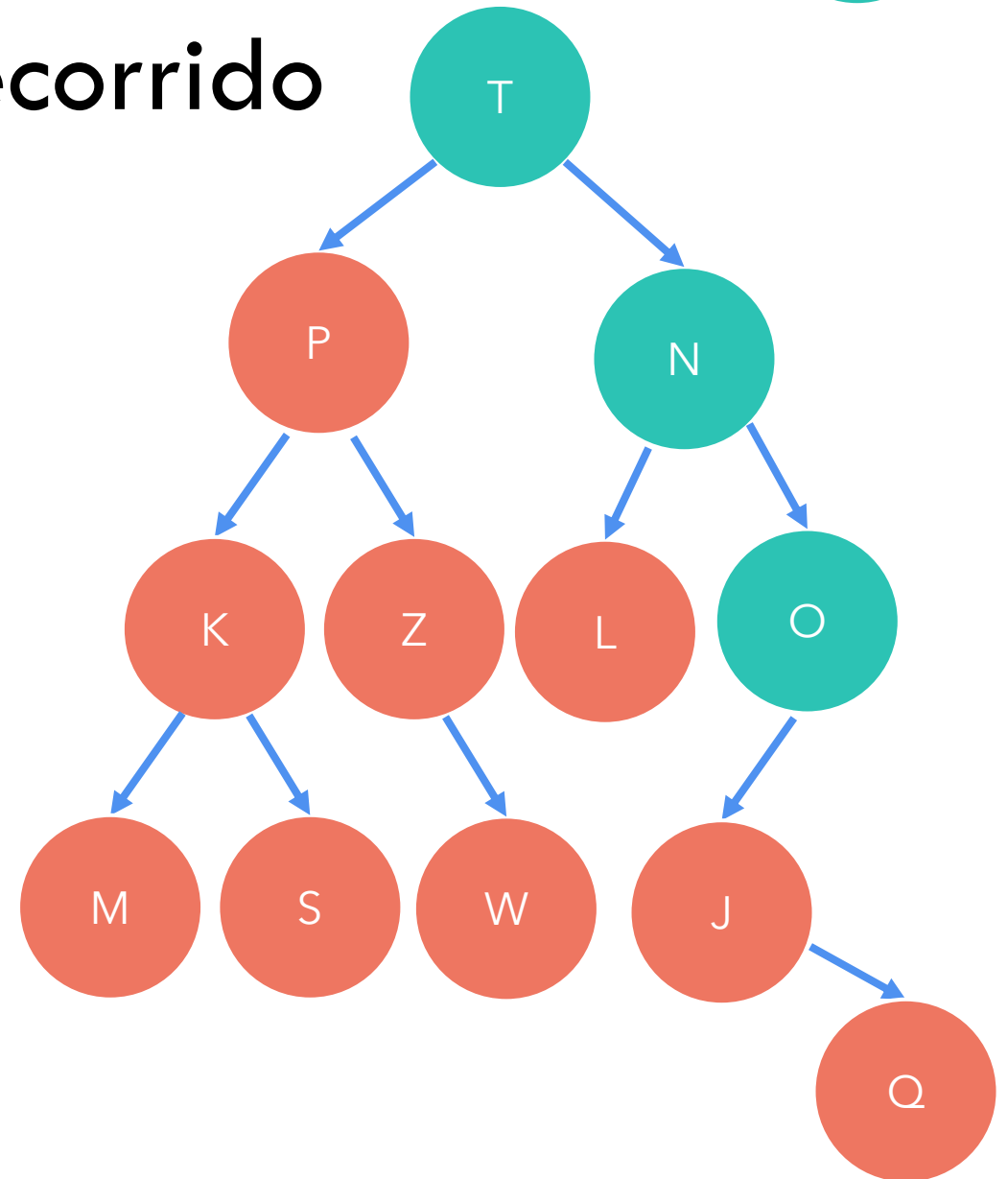
$\text{posorden}(T) = M-S-K-W-Z-P-L-Q$



Arboles – Algoritmos de Recorrido

Algoritmo pos-orden:

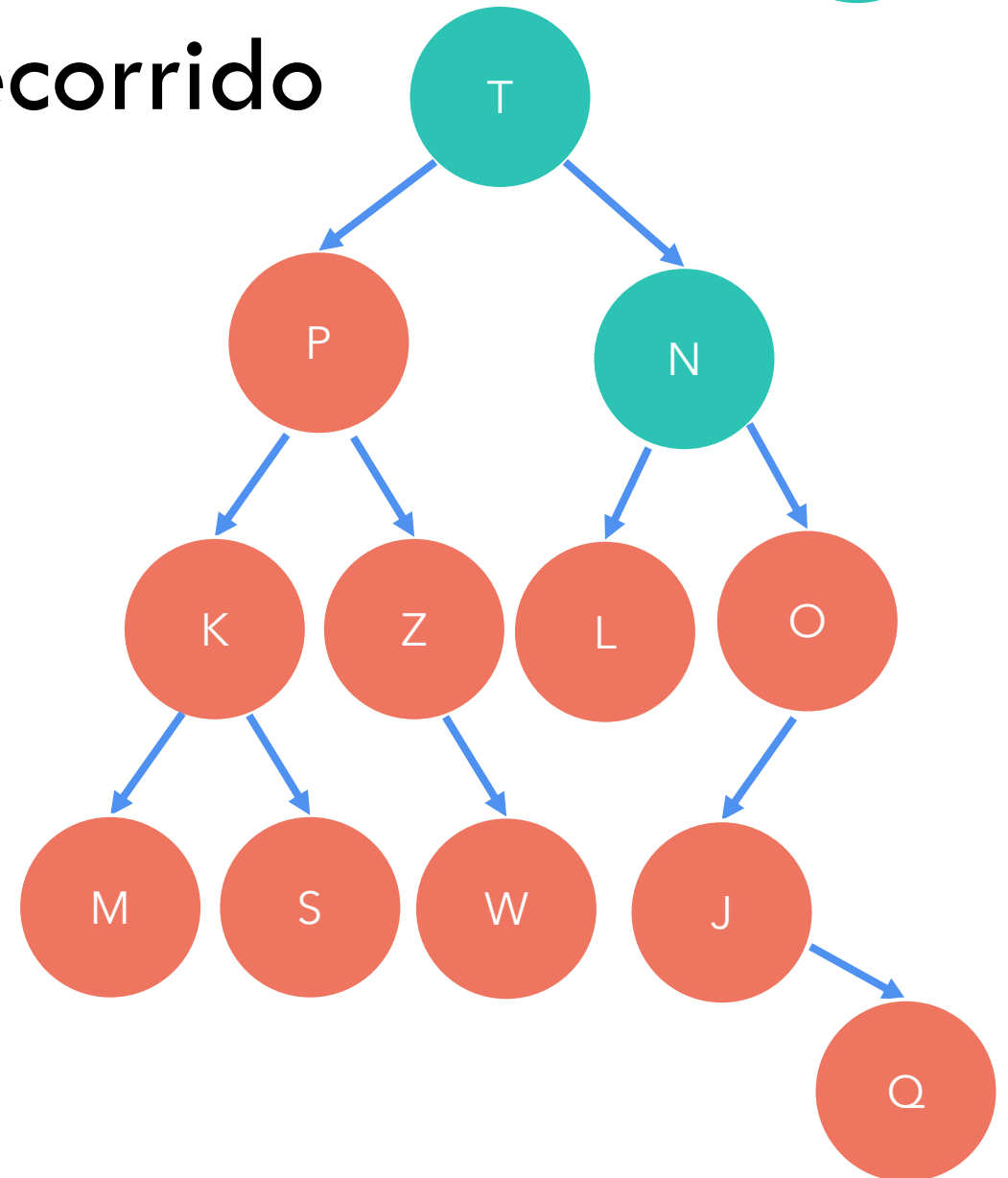
$\text{posorden}(T) = M-S-K-W-Z-P-L-Q-J$



Arboles – Algoritmos de Recorrido

Algoritmo pos-orden:

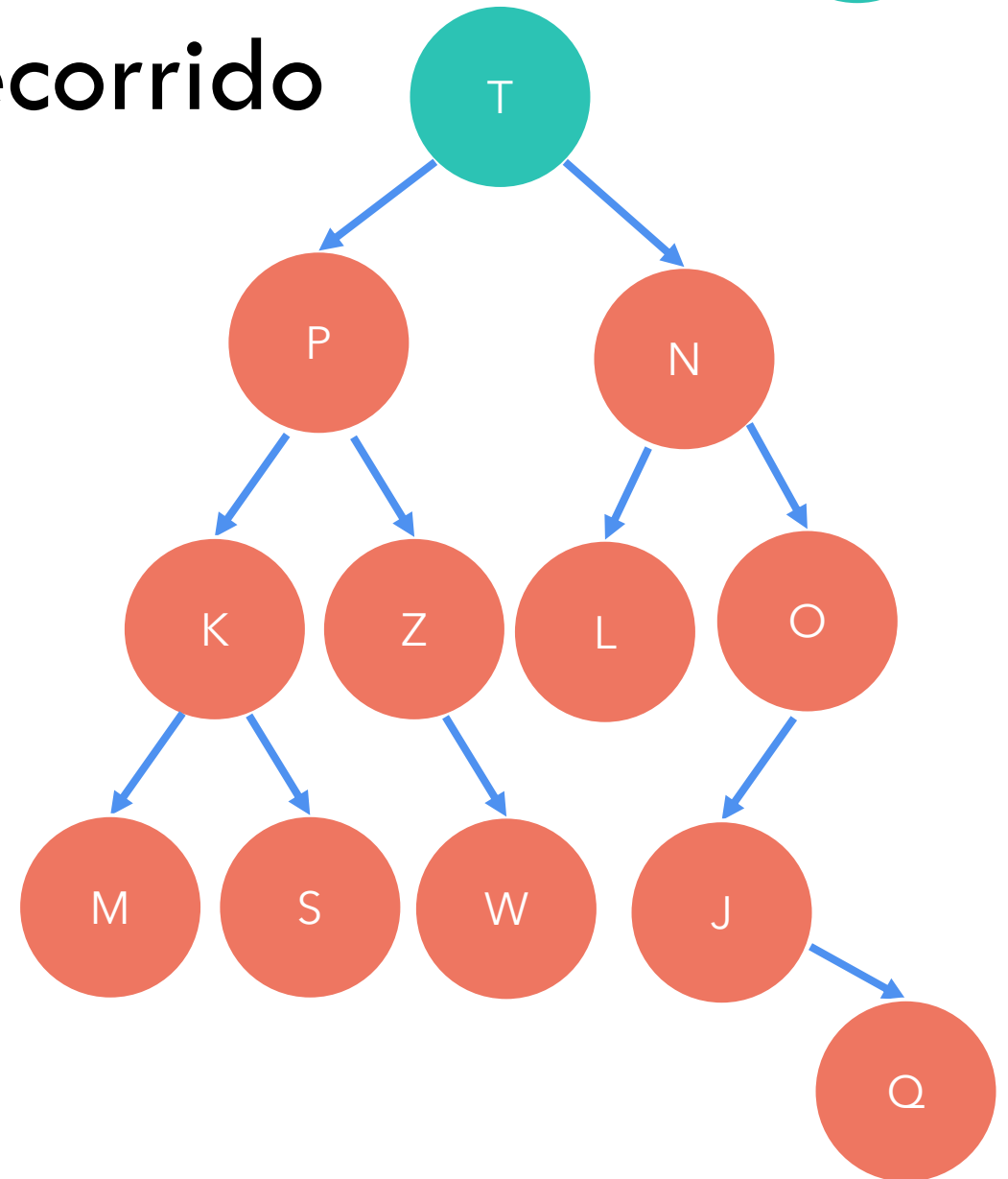
$\text{posorden}(T) = M-S-K-W-Z-P-L-Q-J-O$



Arboles – Algoritmos de Recorrido

Algoritmo pos-orden:

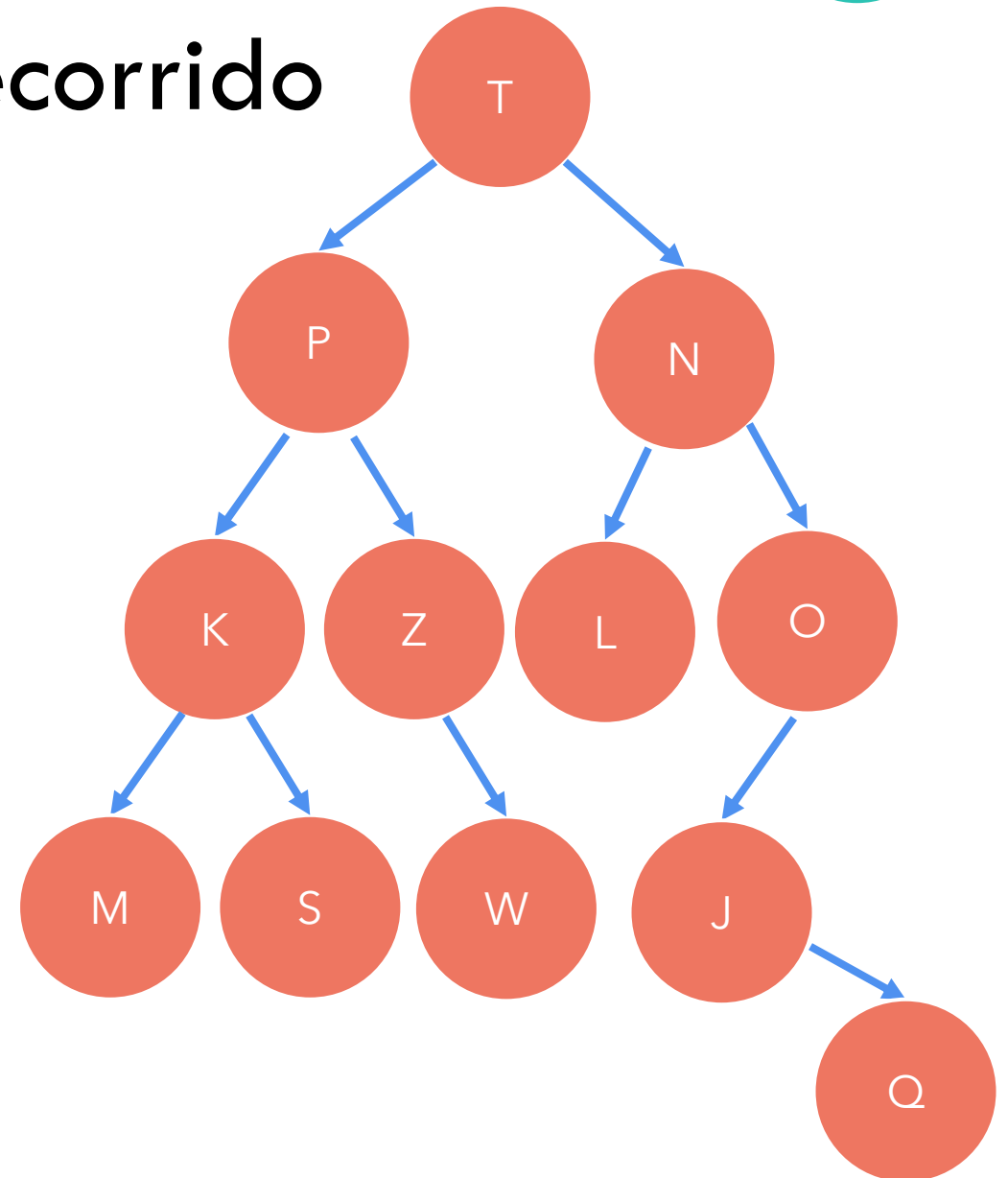
$\text{posorden}(T) = \text{M-S-K-W-Z-P-L-Q-J-O-N}$



Arboles – Algoritmos de Recorrido

Algoritmo pos-orden:

$\text{posorden}(T) = \text{M-S-K-W-Z-P-L-Q-J-O-N-T}$



Arboles – Algoritmos de Recorrido

Algoritmo pos-orden:

Seudocodigo

Posorder(BinaryTree T, Node v)

1. if T.hasLeft(v)
2. Posorder(T,T.left(v))
3. if T.hasRight(v)
4. Posorder(T,T.right(v))
5. visit(v)

El método visitar se puede modificar de acuerdo con el problema. Ejemplos de operación visitar son:

- Un método para imprimir o guardar en un archivo el contenido del nodo
- Acumular los valores numéricos almacenados en el nodo