

MANEJO DE ARBOLES

2023 - CARLOS SOTO

¿Que es recursividad?



- Definición 1: Es una forma de expresar un procedimiento en base a su propia definición, reduciéndolo de manera recurrente a la mínima expresión.
- Definición 2: Ejecutar una tarea o una serie de pasos repetidamente hasta que alcance algún objetivo deseado.
- Definición 3: Es una técnica de programación que busca resolver un problema sustituyéndolo por otros problemas de la misma categoría, pero más simples.
- Definición 4: La idea básica detrás de los algoritmos recursivos, para resolver un problema, resuelve un subproblema que sea una instancia más pequeña del mismo problema, y después usa la solución de esa instancia más pequeña para resolver el problema original.

Ejemplo de recursividad

El factorial de un número se expresa mediante el carácter !. Por ejemplo, factorial de 5 se indica como 5!

El factorial de un número se calcula de la siguiente forma:

Por definición el factorial de 0 es 1:

$$0! = 1$$

Para el resto de números el factorial se calcula multiplicando todos los números desde 1 hasta el propio número.

$$1! = 1$$

$$2! = 2 * 1$$

$$3! = 3 * 2 * 1$$

$$4! = 4 * 3 * 2 * 1$$

En general, el factorial de un número N se calcula multiplicando:

$$N! = N * (N-1) * (N-2) * \dots * 3 * 2 * 1$$

El programa mostrará por pantalla el factorial del número introducido por el usuario.

Si por ejemplo se introduce un 5 el programa mostrará:

$$5! = 120$$

que es el resultado de multiplicar $5 * 4 * 3 * 2 * 1$

Factorial: ↗

Programar un algoritmo recursivo que calcule el factorial de un número.

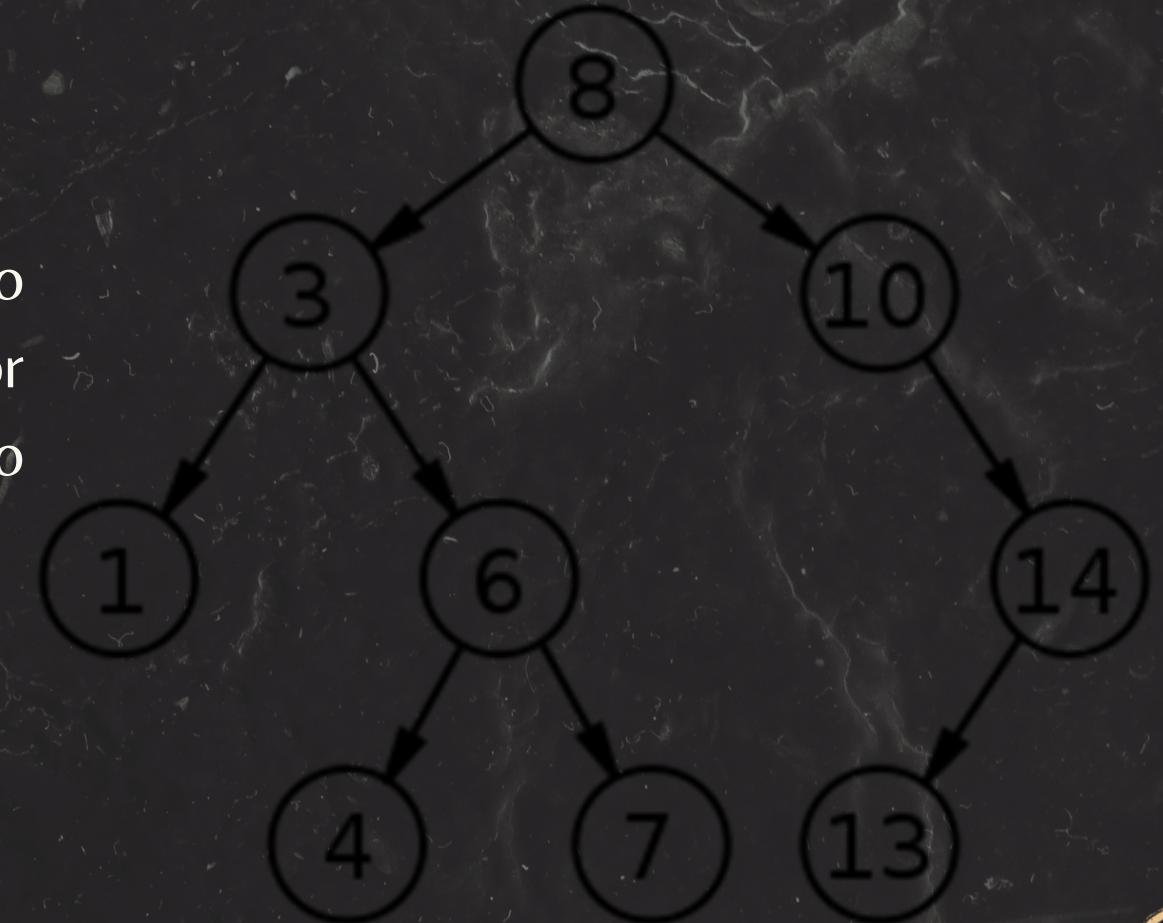
Implementación

Función factorial (Forma 1): ↗

```
public static int factorial(int n){  
    if(n==0) return 1; //AXIOMA  
    else return n*factorial(n-1); //FORMULA RECURSIVA  
}
```

¿Qué es un árbol?

Los árboles son las estructuras de datos más utilizadas, pero también una de las más complejas. Los árboles se caracterizan por almacenar sus nodos en forma jerárquica y no en forma lineal como las Listas ligadas, Colas, pilas, etc.



Árbol Binario de Búsqueda(ABB ó BST)

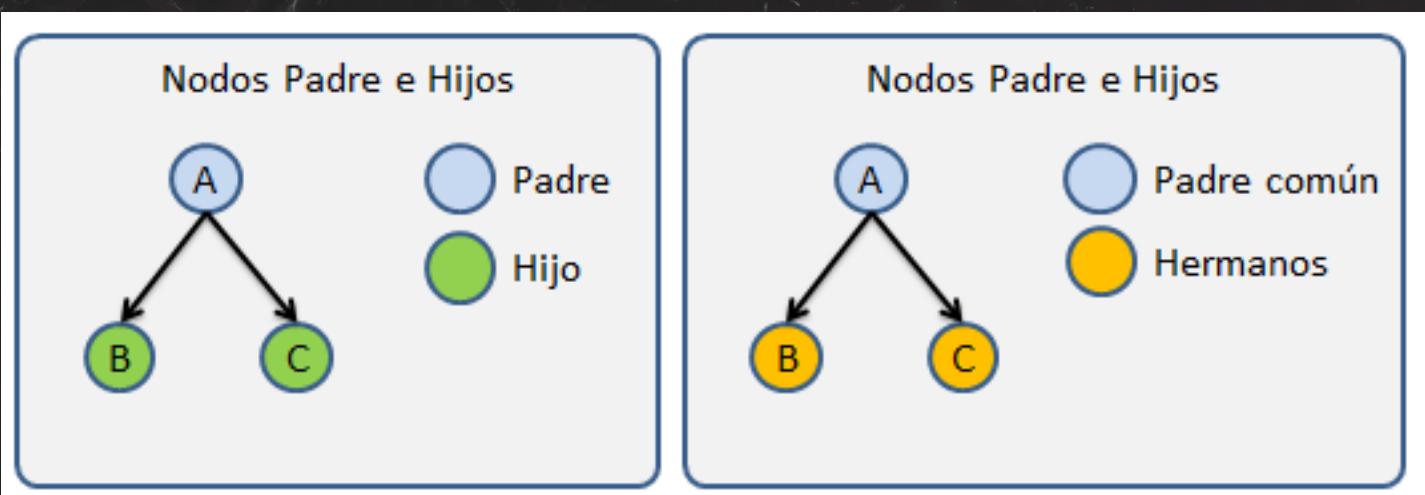
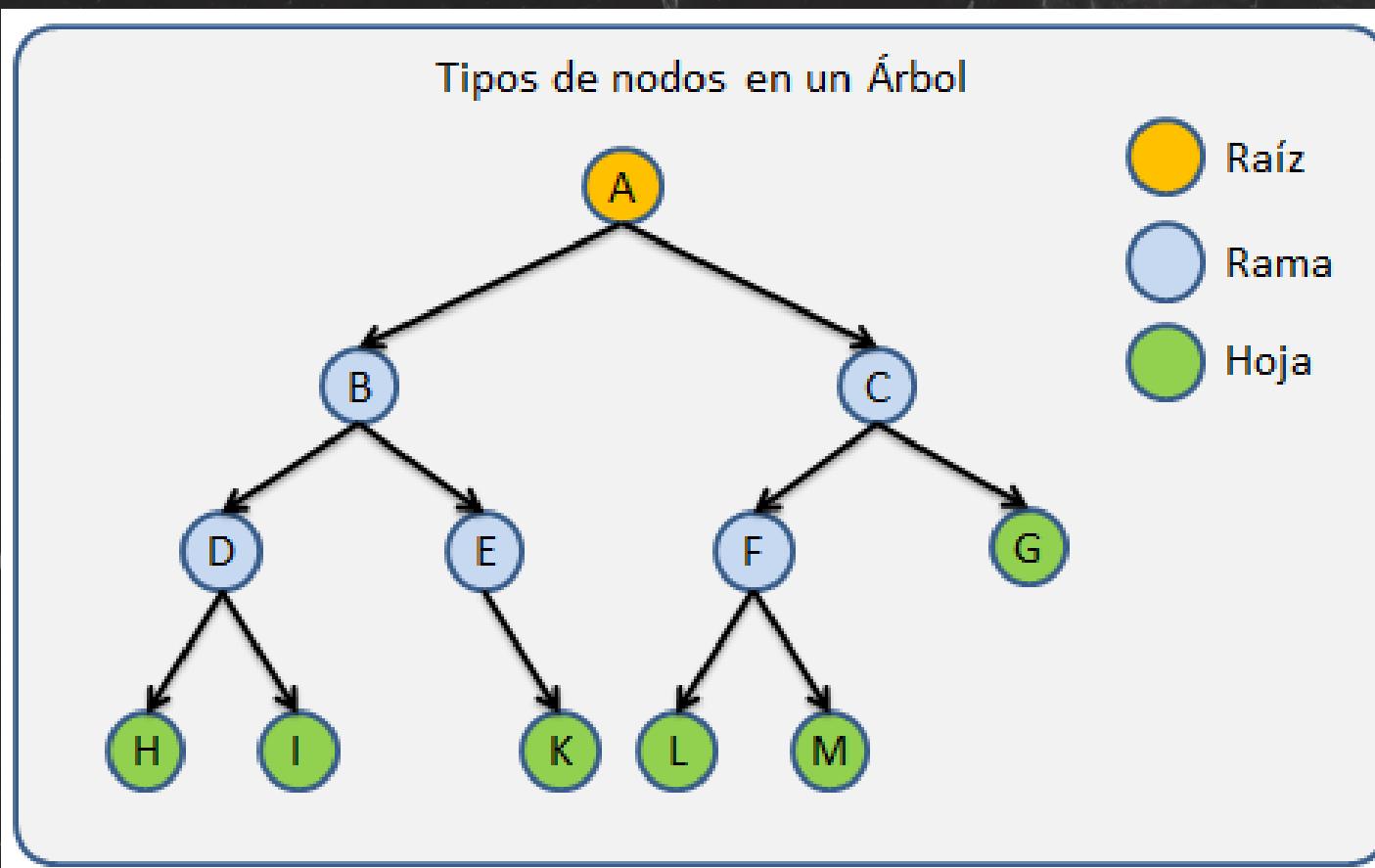


- Es un árbol binario que cumple que el subárbol izquierdo de cualquier nodo (si no está vacío) contiene valores menores que el que contiene dicho nodo, y el subárbol derecho (si no está vacío) contiene valores mayores.
- Es una estructura recursiva.

Datos importantes de los árboles

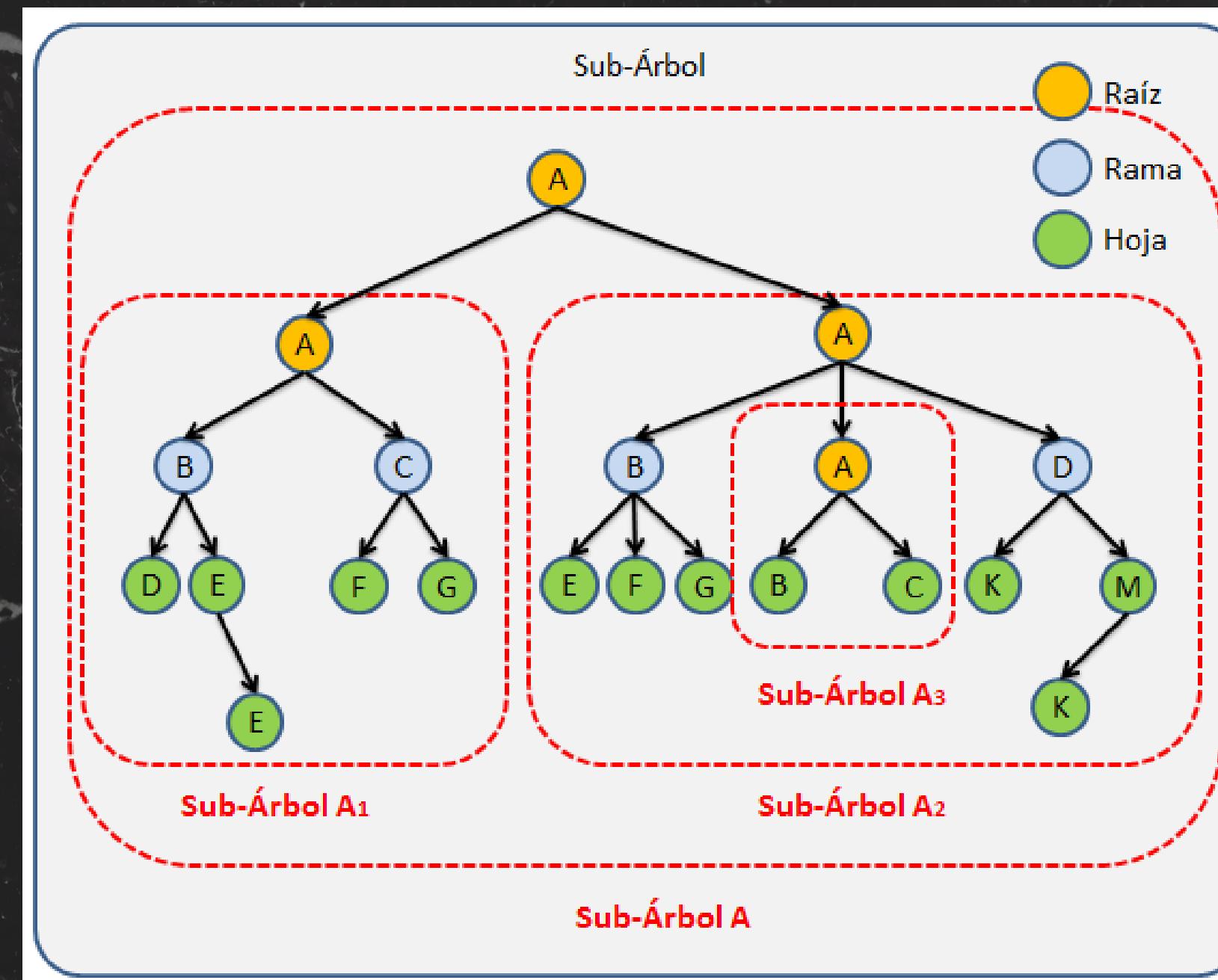
Para comprender de mejor manera que es un árbol comenzaremos explicando como está estructurado.

- Nodos: Se le llama Nodo a cada elemento que contiene un árbol.
- Nodo Raíz: Se refiere al primer nodo de un árbol, solo un nodo del árbol puede ser la raíz.
- Nodo Padre: Se utiliza este término para llamar a todos aquellos nodos que tienen al menos un hijo.
- Nodos hijo: Los hijos son todos aquellos nodos que tiene un padre.
- Nodo Hermano: Los nodos hermanos son aquellos nodos que comparte a un mismo parente en común dentro de la estructura.
- Nodo Hoja: Son todos aquellos nodos que no tienen hijos, los cuales siempre se encuentran en los extremos de la estructura.
- Nodo Rama: Estos son todos aquellos nodos que no son la raíz y que además tiene al menos un hijo.



Sub-Árbol

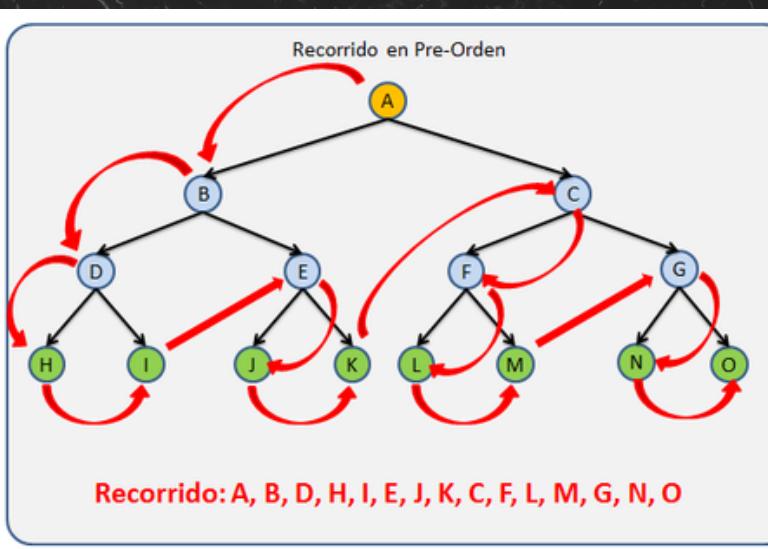
Conocemos como Sub-Árbol a todo Árbol generado a partir de una sección determinada del Árbol, Por lo que podemos decir que un Árbol es un nodo Raíz con N Sub-Árboles.



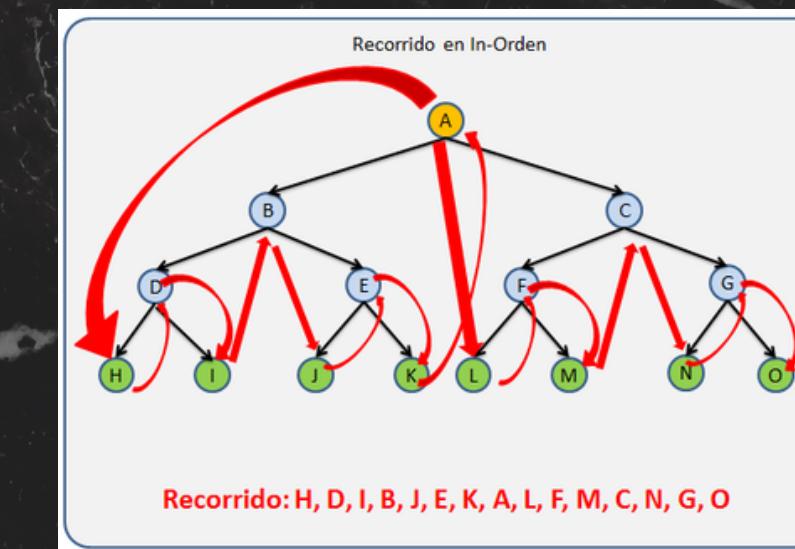
Recorrido sobre Árboles:

Las búsquedas no informadas son aquellas en que se realiza el viaje por todo el árbol sin tener una pista de donde pueda estar el dato deseado. Este tipo de búsquedas también se conocen como búsquedas a ciegas.

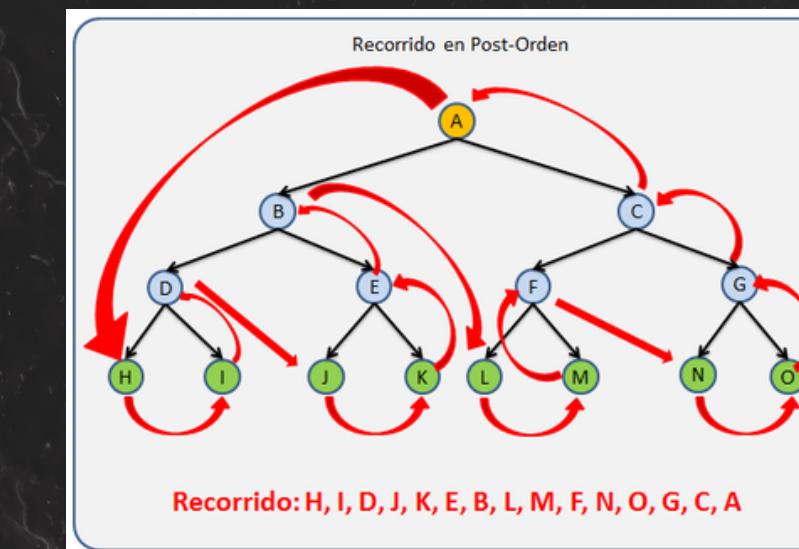
El recorrido de un arbol es el proceso para recorrer (desplazarse a lo largo) un arbol de manera sistematica a fin de que cada vertice se visite y procese exactamente una vez. Hay tres metodos para recorrer un arbol binario a saber recorridos de preorden , de inorder y de posorden.



PREORDEN



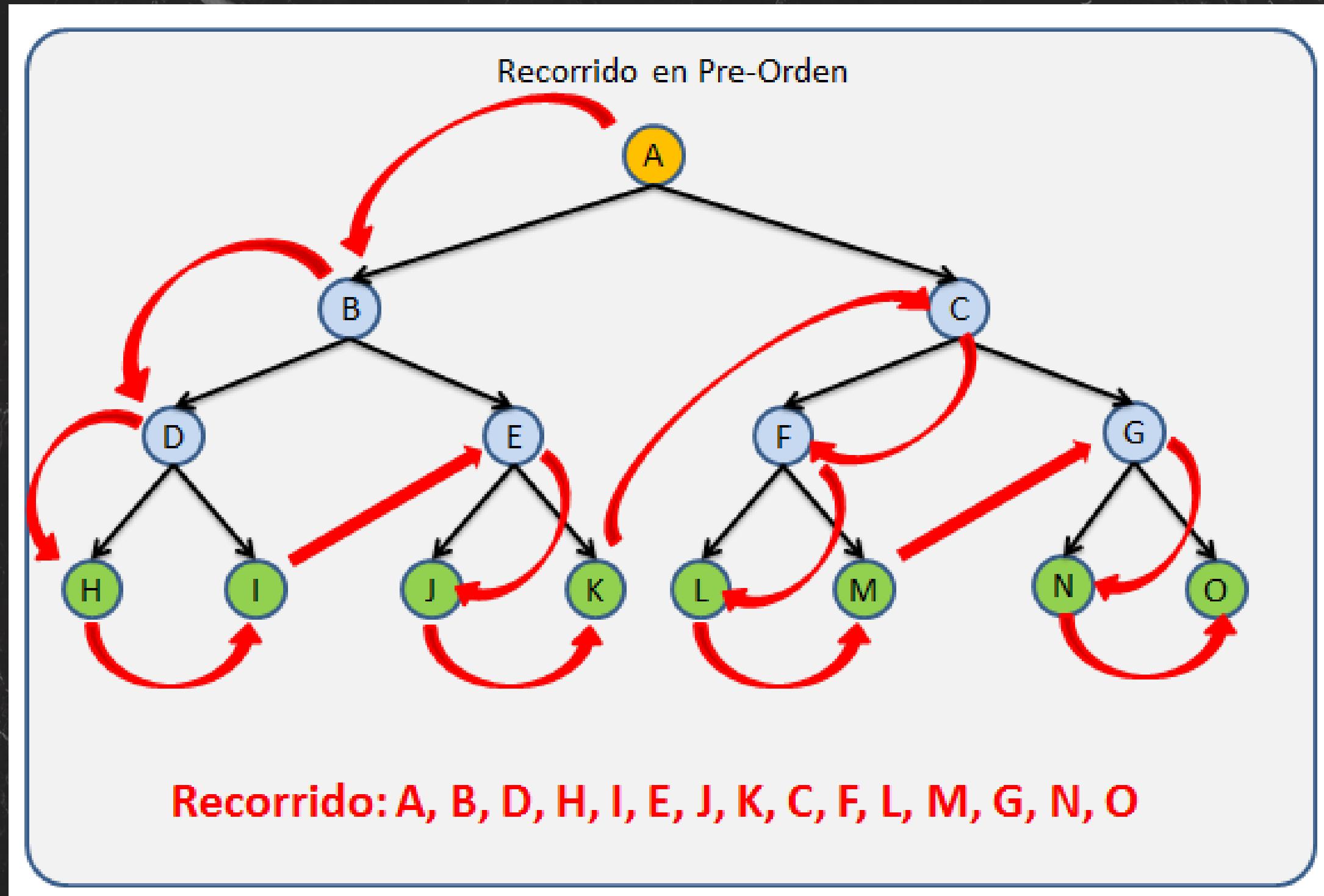
INORDEN



POSTORDEN

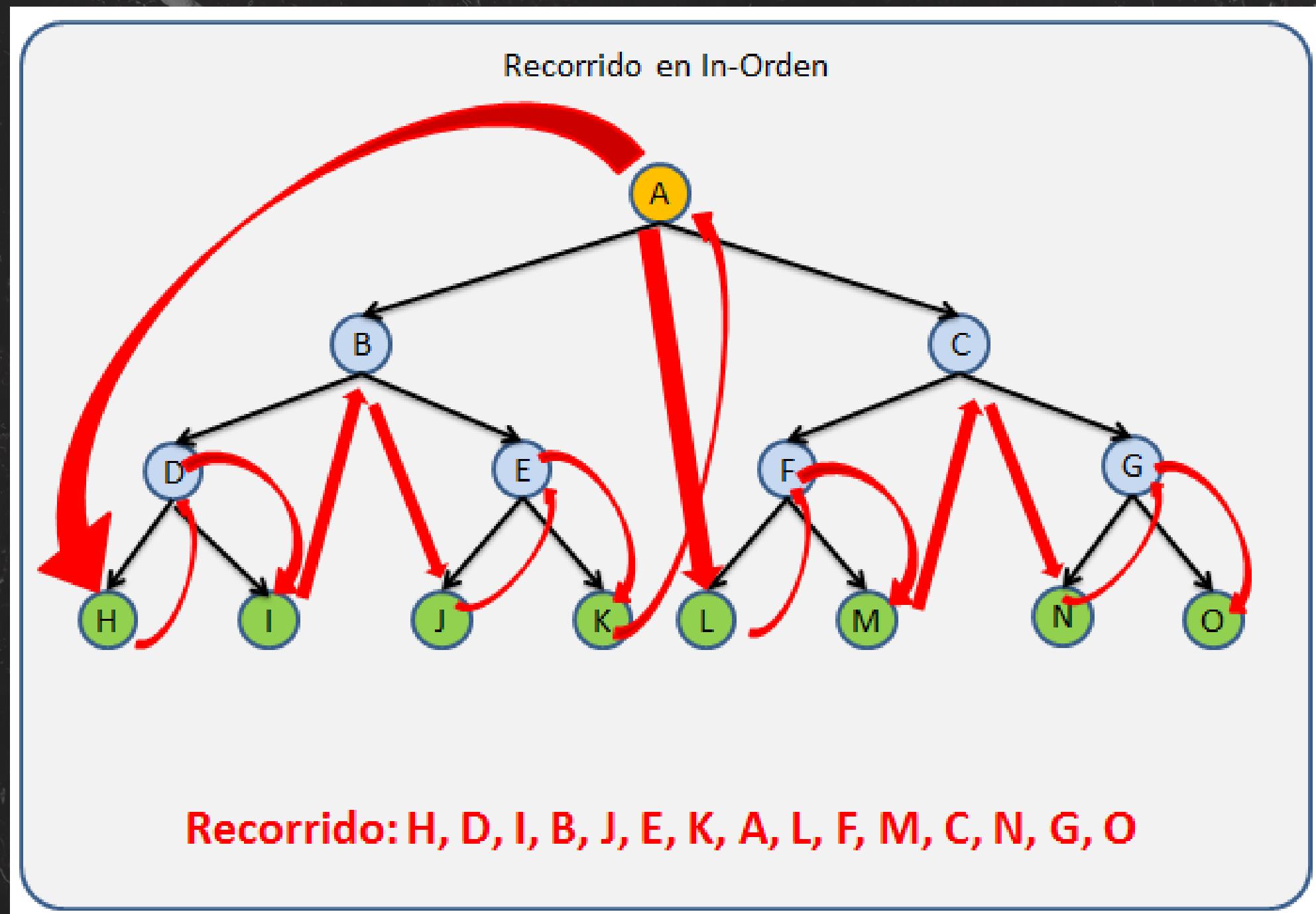
Recorrido Pre-orden

El recorrido inicia en la Raíz y luego se recorre en pre-orden cada uno de los sub-árboles de izquierda a derecha.



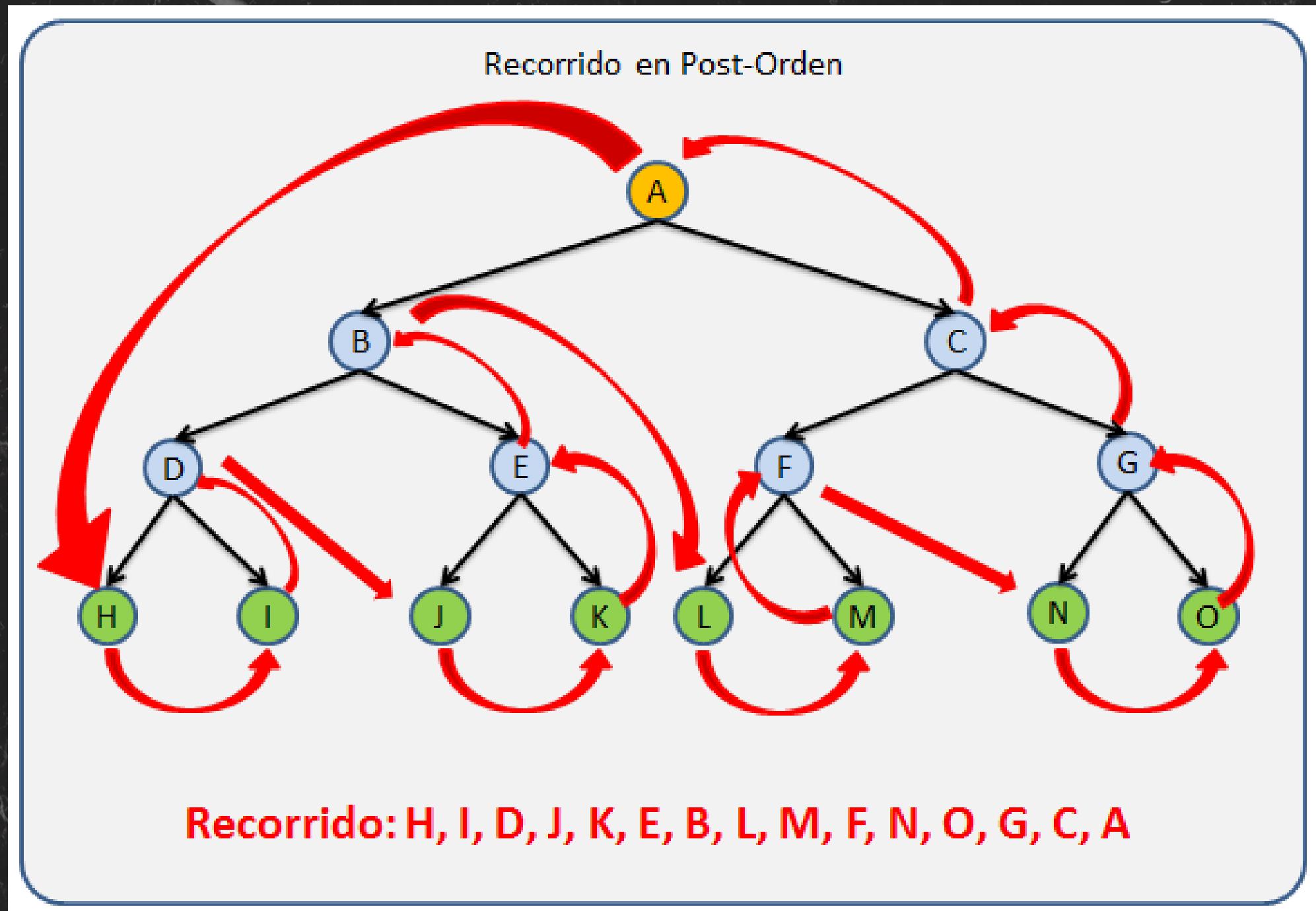
Recorrido In-orden

Se recorre en in-orden el primer sub-árbol, luego se recorre la raíz y al final se recorre en in-orden los demás sub-árboles.



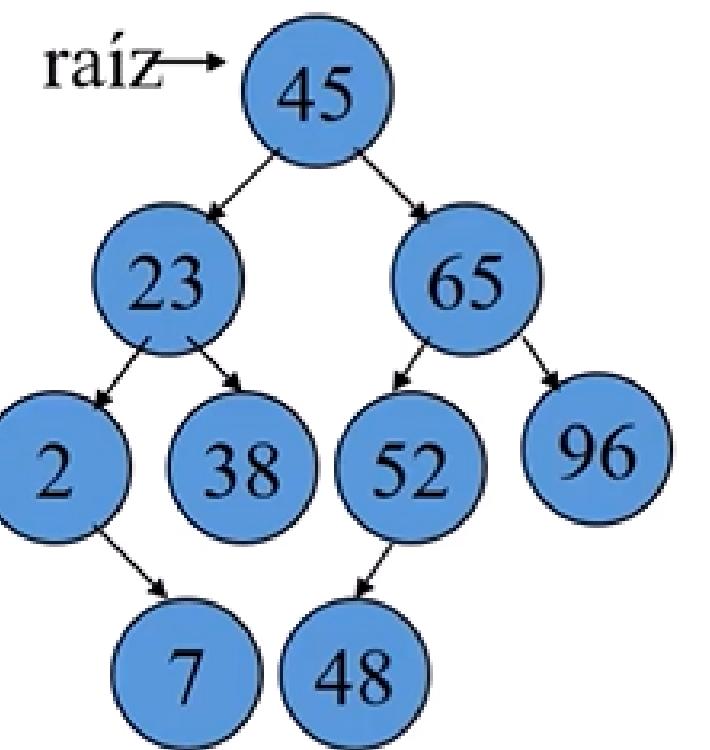
Recorrido Post-orden

Se recorre el pos-orden cada uno de los sub-árboles y al final se recorre la raíz.



Ejemplo práctico

Recorridos



- **Inorden = 2,7,23,38,45,48,52,65,96**
 - Visitar el subarbol izquierdo en inorden
 - Visitar el nodo raiz
 - Visitar el subarbol derecho en inorden
- **Preorden= 45,23,2,7,38,65,52,48,96**
 - Visitar el nodo raiz
 - Visitar el subarbol izquierdo en preorden
 - Visitar el subarbol derecho en preorden
- **Postorden= 7,2,38,23,48,52,96,65,45**
 - Visitar el subarbol izquierdo en postorden
 - Visitar el subarbol derecho en postorden
 - Visitar el nodo raiz

GRACIAS POR SU
ATENCION

