UNIDAD III
Estructuras de Datos Jerárquicas

Estructura de Datos

Generalidades

- Son estructuras compejas no lineales y dinámicas donde en su ejecución varia el número de elementos y uso de memoria a lo largo del programa.

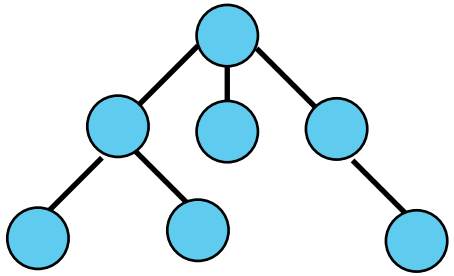
- Entre este tipo de estructuras se pueden mencionar por ejemplo: árboles, grafos y redes.

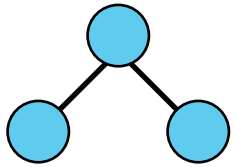
Árboles

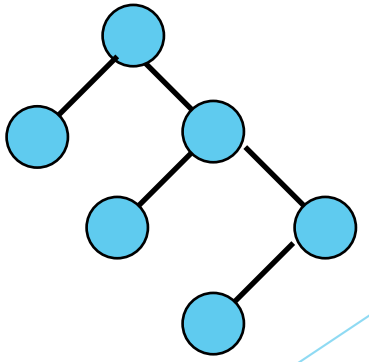
Estructura de Datos

Definición

- Son Estructuras de Datos no lineales.
- Es una colección de nodos donde cada uno, además de almacenar información, guarda la dirección de sus sucesores.
- Es una estructura de datos jerárquica.
- La relación entre los elementos es de uno a muchos.

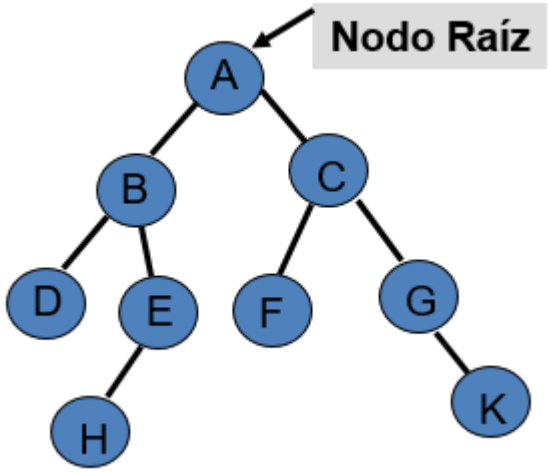






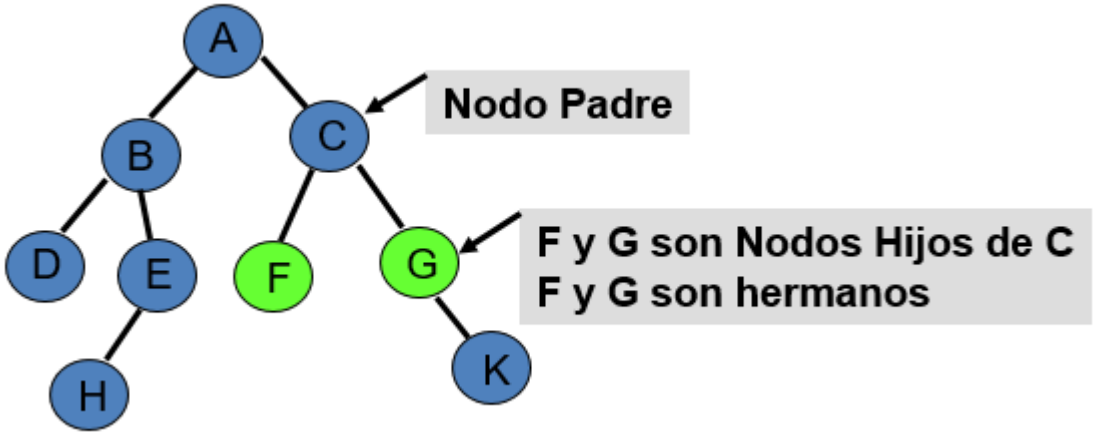
Terminología

- **Nodo:** Cada elemento en un árbol.
- **Nodo Raíz:** Primer elemento agregado al árbol.



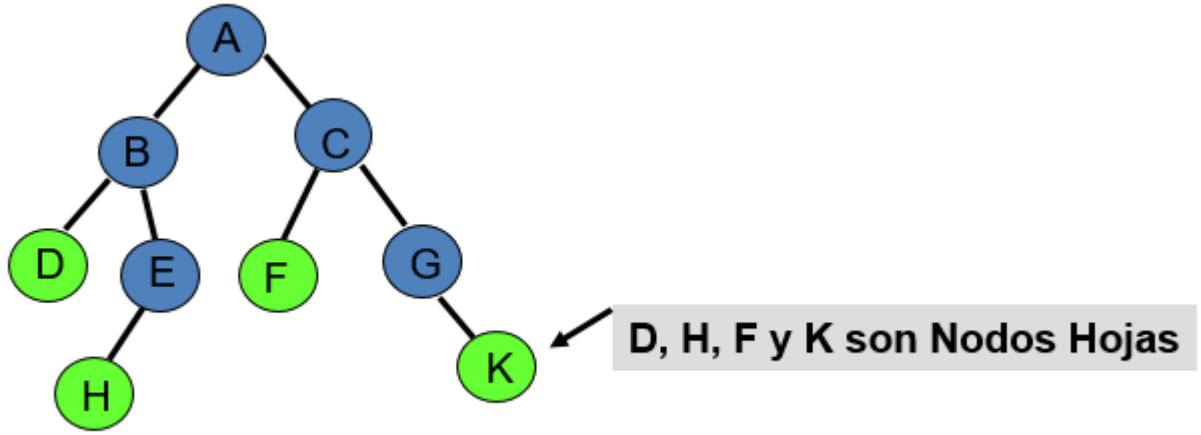
Terminología

- **Nodo Padre:** Se le llama así al nodo predecesor de un elemento.
- **Nodo Hijo:** Es el nodo sucesor de un elemento.
- **Hermanos:** Nodos que tienen el mismo nodo padre.



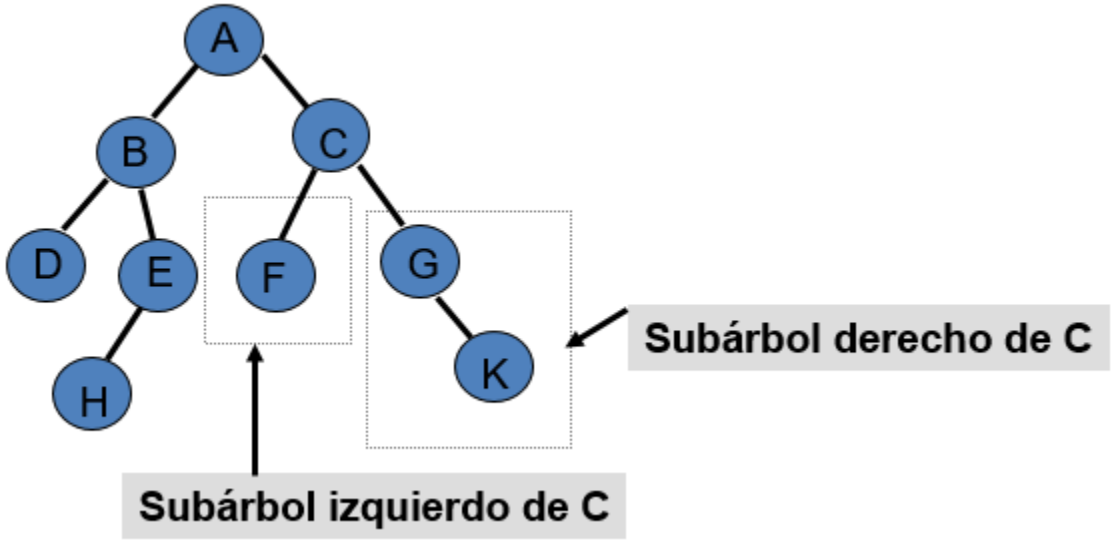
Terminología

- **Nodo Hoja:** Aquel nodo que no tiene hijos.

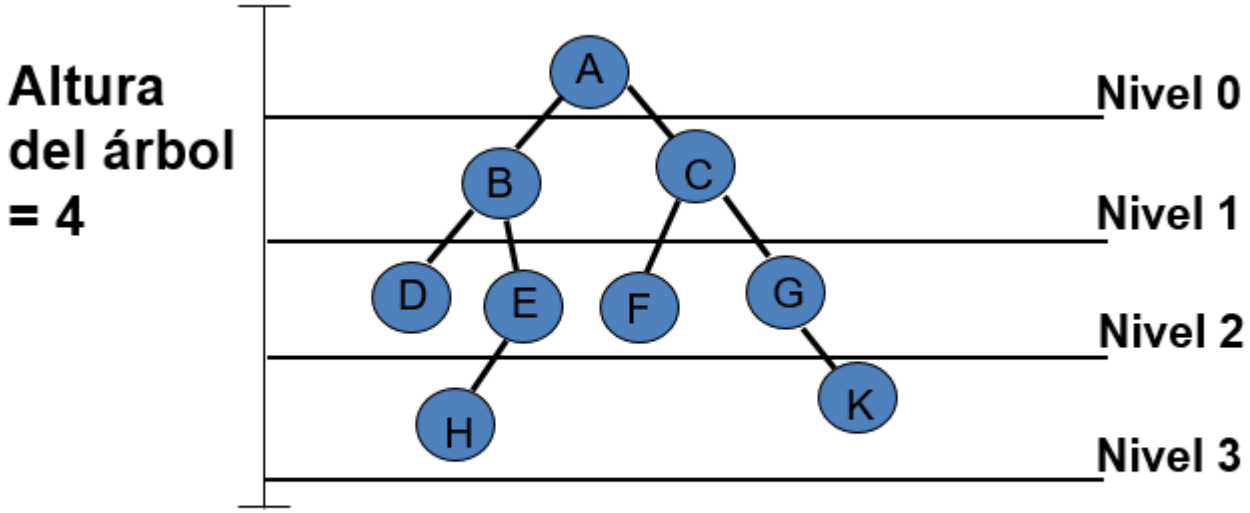


Terminología

- **Subárbol:** Todos los nodos descendientes por la izquierda o derecha de un nodo.



Altura y niveles

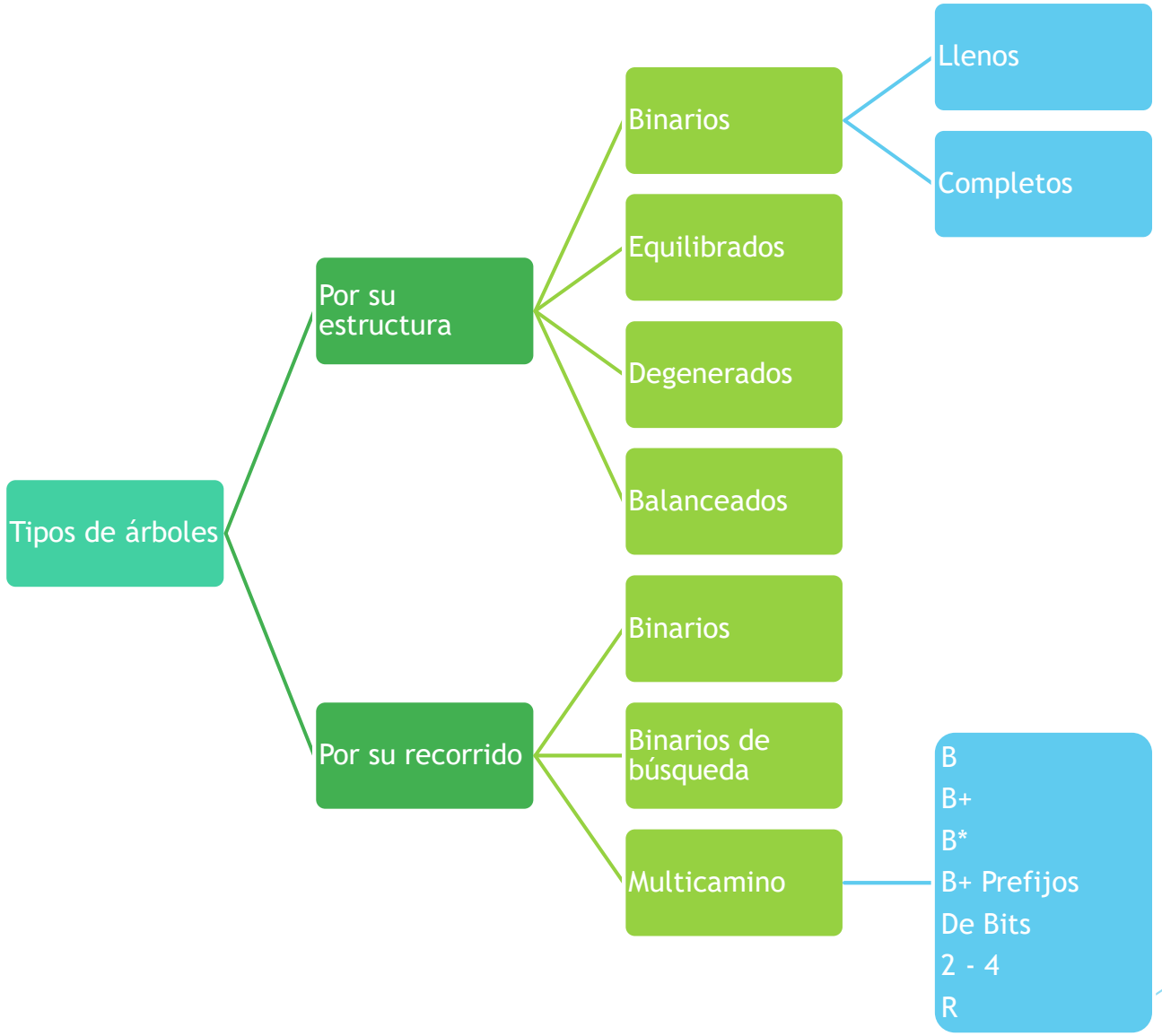


La Altura es la cantidad de niveles.

Elementos del árbol

- Nodo Raíz, Nodos Hijos, Nodos Hermanos, Altura, Recorridos, Dirección.
- Todo Árbol tiene un solo Nodo Raíz.
- Los Árboles pueden tener o no Nodos Hijos. En caso de tenerlos, pueden existir Nodos Hermanos.
- Si únicamente tiene un Nodo Raíz, su Altura = 1 y su Nivel = 0.
- Los Recorridos pueden ser en preorden, postorden y en orden.
- Un Árbol puede recorrerse en dirección Top-Down de arriba abajo, de abajo arriba, (Down-Top).
- A partir de su rama Izquierda o a partir de su rama Derecha.

Tipos de árbol



Recorrido de árboles

Estructura de Datos

Recorrido de árboles

- Es el proceso de visitar de una manera sistemática, exactamente una vez, cada nodo en una estructura de datos de árbol (examinando y/o actualizando los datos en los nodos).

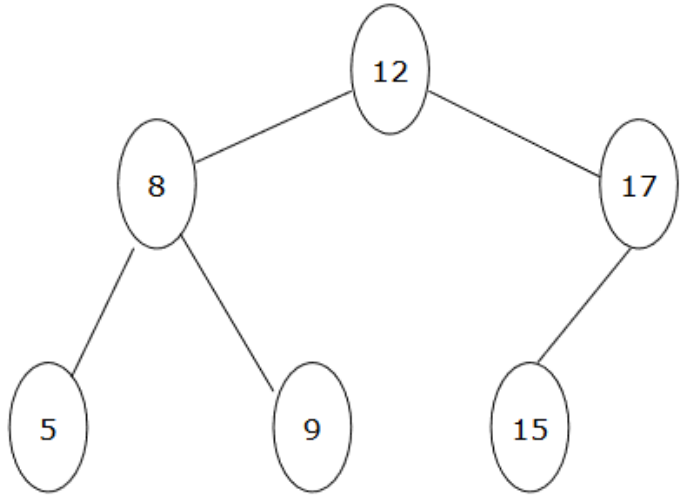
- Tales recorridos están clasificados por el orden en el cual son visitados los nodos.

- Pueden ser:
  - Recorrido en amplitud
  - Recorrdido en profundidad

Recorrido en amplitud

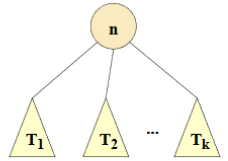
- Es aquel recorrido que recorre el árbol por niveles del nivel superior a los niveles inferiores

12, 8, 17, 5, 9, 15.



Recorrido en profundidad

- La ordenación o recorrido de árboles suele hacerse de tres modos:
  - **Preorden:** la raíz se recorre antes que los recorridos de los subárboles izquierdo y derecho
  - **Inorden:** la raíz se recorre entre los recorridos de los árboles izquierdo y derecho
  - **Postorden:** la raíz se recorre después de los recorridos por el subárbol izquierdo y el derecho
- Preorden (antes), inorden (en medio), postorden (después)



Preorden: $n,T_1,T_2,...,T_k$

Postorden: $T_1,T_2,...,T_k,n$

Inorden: $T_1,n,T_2,...,T_k$

Figura A

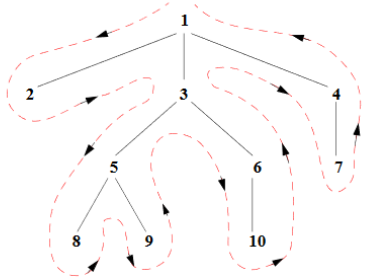


Figura B

Recorrido preorden

Preorden: (raíz, izquierdo, derecho)

Para recorrer un árbol binario no vacío en preorden, hay que realizar las siguientes operaciones recursivamente en cada nodo, comenzando con el nodo de raíz:

1. Visitar la raíz
2. Atravesar el sub-árbol izquierdo
3. Atravesar el sub-árbol derecho

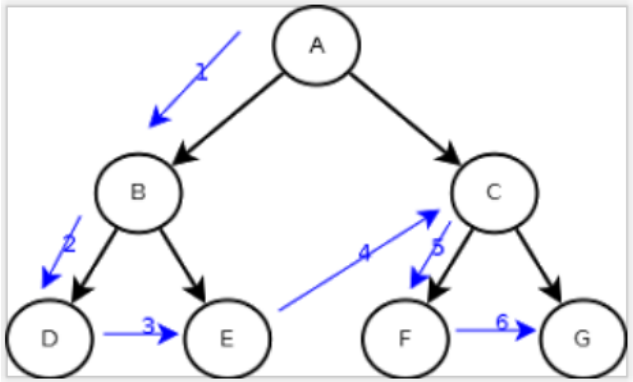
```
método Preorden (N : Nudo; A : Arbol)
  listar N;
  para cada hijo H de N, y empezando por la izquierda
  hacer
    Preorden(H,A);
  fpara;
fmétodo;
```

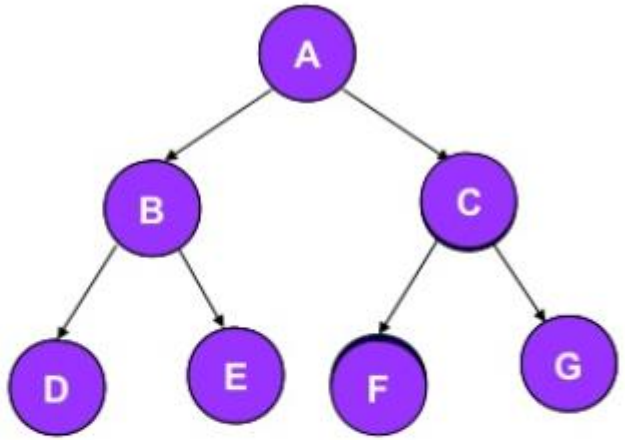
Implementación (preorden)

```
void preorden(tArbol *a)
{
  if (a != NULL) {
    tratar(a);                          //Realiza una operación en nodo
    preorden(a->hIzquierdo);
    preorden(a->hDerecho);
  }
}
```

```
push(s,NULL);       //insertamos en una pila (stack) el valor NULL, para asegurarnos de que esté vacía
push(s,raíz);                       //insertamos el nodo raíz
MIENTRAS (s <> NULL) HACER
    p = pop(s);                     //sacamos un elemento de la pila
    tratar(p);                      //realizamos operaciones sobre el nodo p
    SI (D(p) <> NULL)               //preguntamos si p tiene árbol derecho
         ENTONCES push(s,D(p));
    FIN-SI
    SI (I(p) <> NULL)               //preguntamos si p tiene árbol izquierdo
         ENTONCES push(s,I(p));
    FIN-SI
FIN-MIENTRAS
```

Ejemplo (preorden)

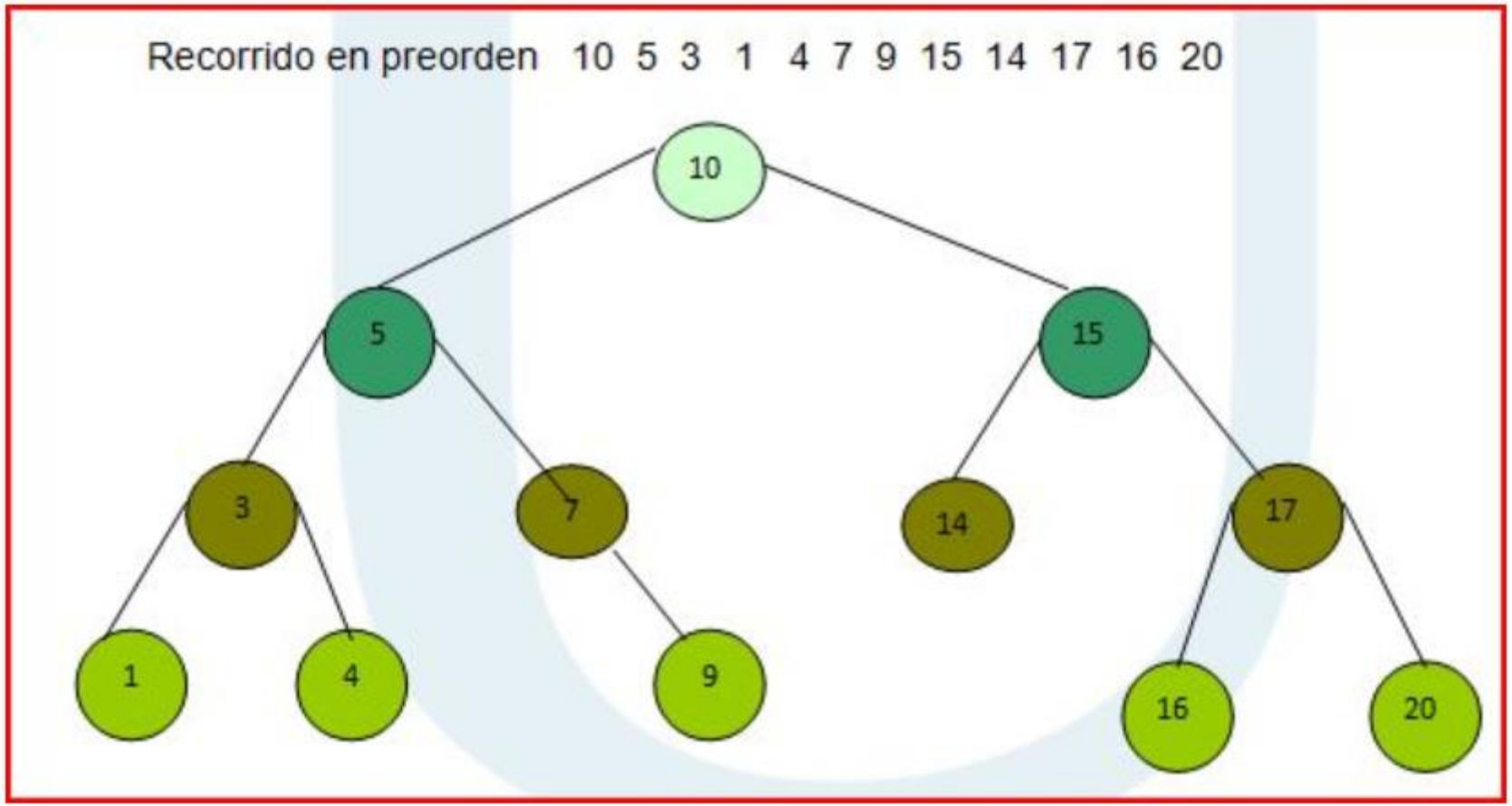




Recorrido: A B D E C F G

1. Raíz.
2. Subárbol izquierdo en preorden.
3. Subárbol derecho en preorden.

Ejemplo (preorden)



Recorrido inorden

Inorden: (izquierdo, raíz, derecho).

Para recorrer un árbol binario no vacío en inorden (simétrico), hay que realizar las siguientes operaciones recursivamente en cada nodo:

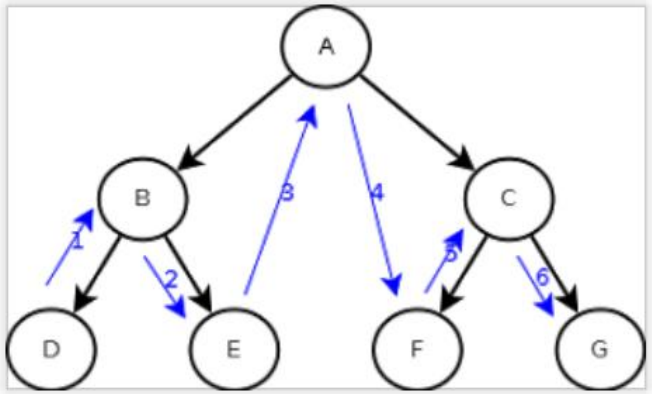
1. Atravesar el sub-árbol izquierdo
2. Visitar la raíz
3. Atravesar el sub-árbol derecho

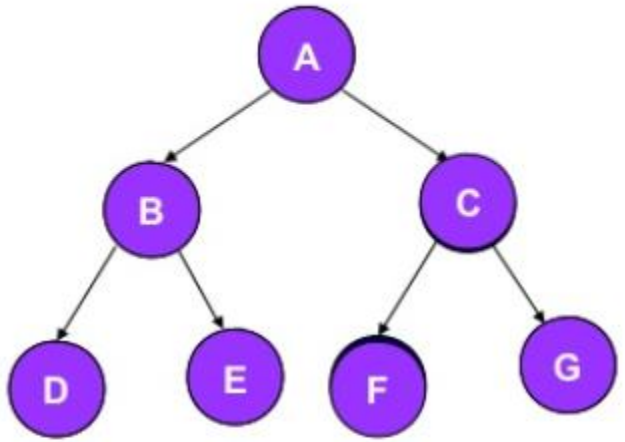
```
método Inorden (N : Nudo; A : Arbol)
  si n es una hoja entonces
    listar n;
  si no
    Inorden(hijo más a la izquierda de n,A);
    listar n;
    para cada hijo h de n, excepto el más a la
        izquierda, y empezando por la izquierda
    hacer
      Inorden(H,A);
    fpara;
  fsi;
fmétodo;
```

Implementación (inorden)

```
void inorden(tArbol *a)
{
  if (a != NULL) {
    inorden(a->hIzquierdo);
    tratar(a);                                  //Realiza una operación en nodo
    inorden(a->hDerecho);
  }
}
```

Ejemplo (inoden)

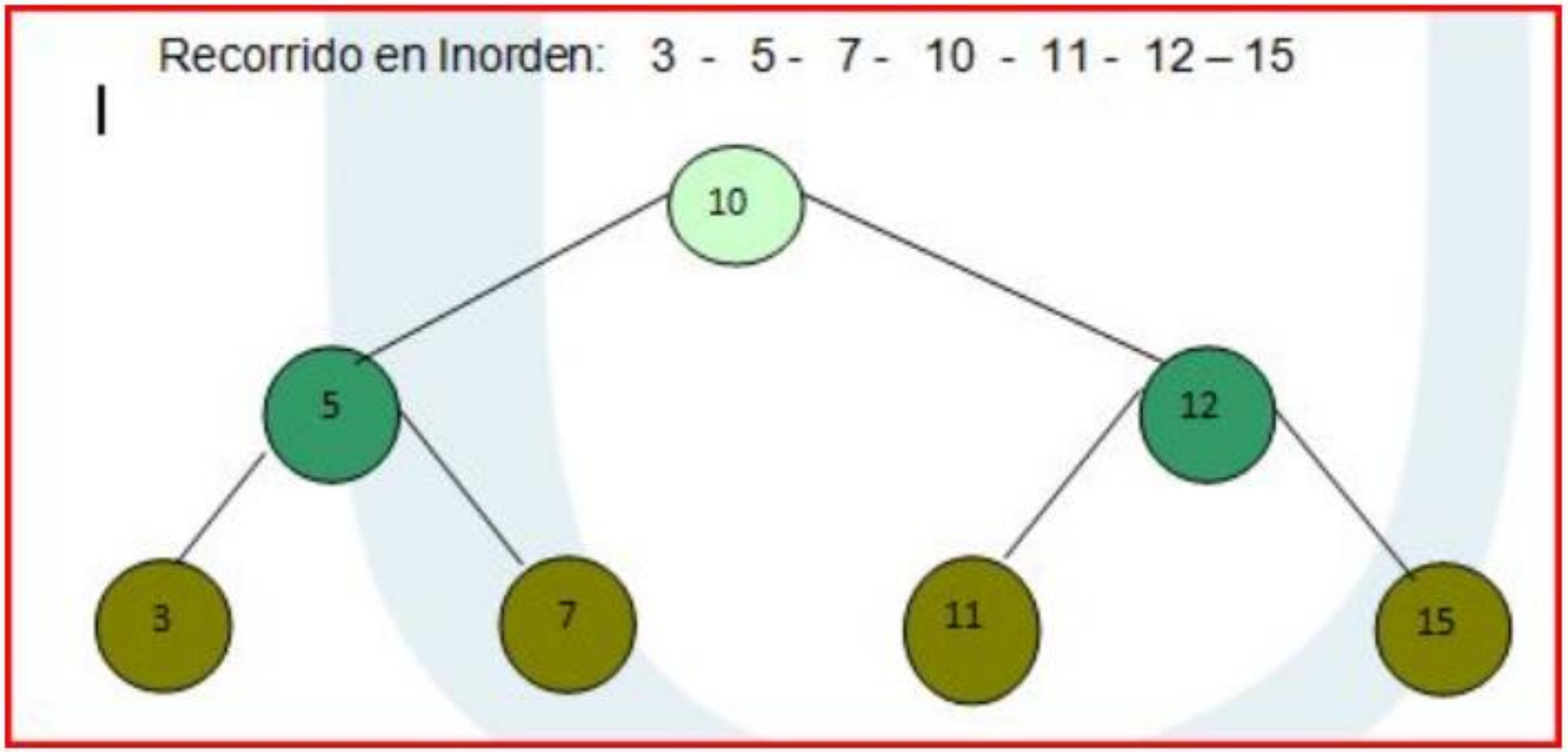




Recorrido

1. Subárbol izquierdo en simétrico.
2. Raíz.
3. Subárbol derecho en simétrico.

Ejemplo (inoden)



Recorrido postorden

Postorden: (izquierdo, derecho, raíz).

Para recorrer un árbol binario no vacío en postorden, hay que realizar las siguientes operaciones recursivamente en cada nodo:

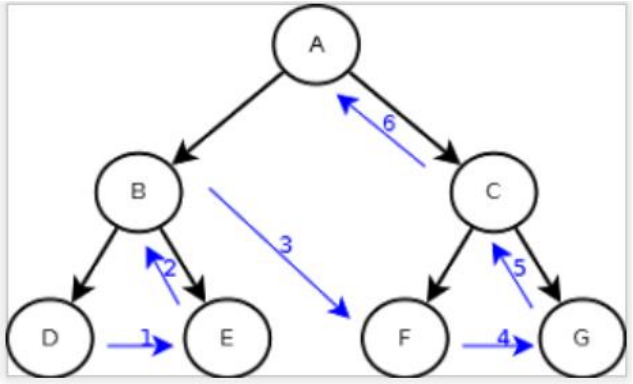
1. Atravesar el sub-árbol izquierdo
2. Atravesar el sub-árbol derecho
3. Visitar la raíz

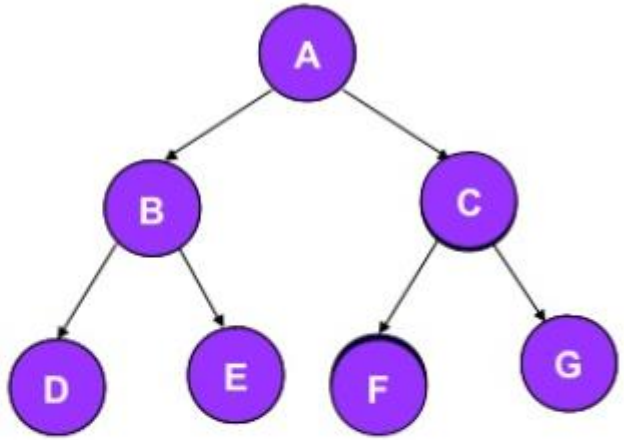
```
método Postorden (N : Nudo; A : Arbol)
  para cada hijo H de N, y empezando por la izquierda
  hacer
    Postorden(H,A);
  fpara;
  listar N;
fmétodo;
```

Implentación (postorden)

```
void postorden(tArbol *a)
{
  if (a != NULL) {
    postorden(a->hIzquiedo);
    postorden(a->hDerecho);
    tratar(a);                          //Realiza una operación en nodo
  }
}
```

Ejemplo (postorden)



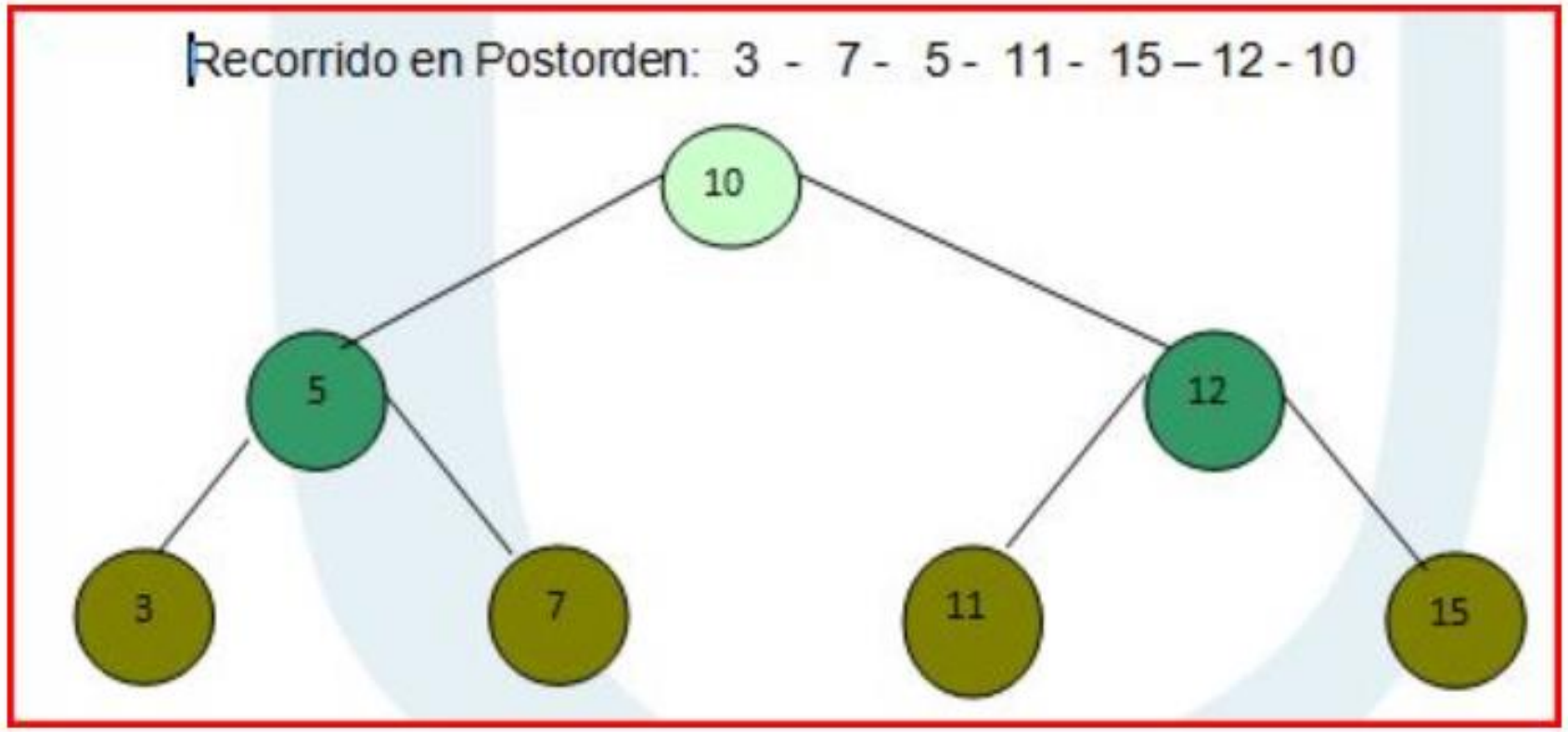


D E B F G C A

Recorrido

1. Subárbol izquierdo en orden final.
2. Subárbol derecho en orden final.
3. Raíz.

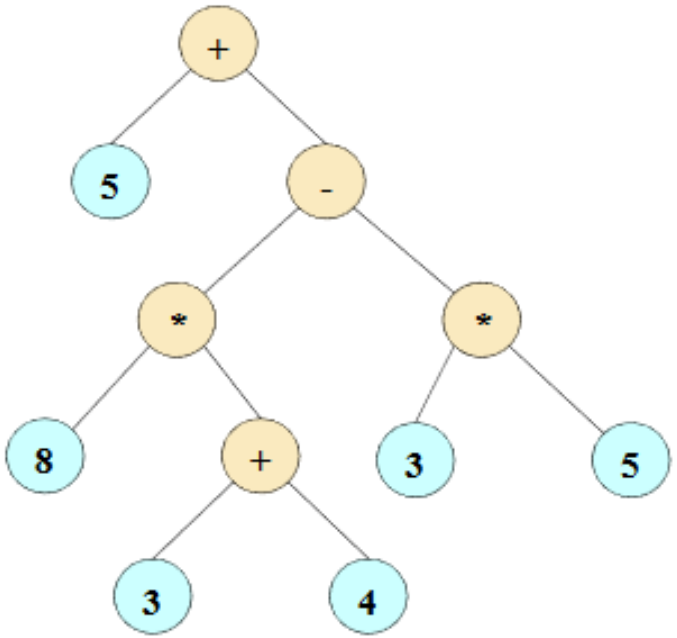
Ejemplo (postorden)



Ejemplo de ordenación de expresiones aritméticas

Expresión: 5+8*(3+4)-3*5:

- ***preorden***: **+5-*8+3,4*3,5**
- ***inorden***: **5+(8*(3+4)-(3*5)) es la expresión en notación matemática normal**
- ***postorden***: **5,8,3,4+*3,5*-+ es la expresión en Notación Polaca Inversa (RPN)**



GRACIAS