

Grado → grado del árbol = nodo c. > grado

Altura → es la long. del camino + largo desde el nodo hasta una hoja.

↳ Altura de un árbol es la h del nodo raíz

↳ Tiene h=0

Profundidad/nivel → long. del único camino de la raíz hasta el nodo

↳ Tiene prof/nivel 0

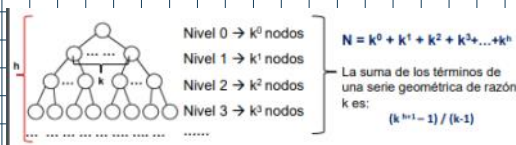
A. lleno → Dado árbol T de grado k y alt. H, T. es lleno si cada

nodo interno tiene grado k y los hijos están al mismo nivel

Si T es lleno, con. de nodos es $(k^{h+1}-1)/(k-1)$

A. completo → Si el lleno hasta h-1 y h se completa de izquierda a derecha.

Si T es completo, con. de $(k^h + k - 2)/(k - 1) > (k^{h+1} - 1)/(k - 1)$



lleno



completo

BUSCAR POR QUE SON ASI

➤ Preorden

Se procesa primero la raíz y luego los hijos

➤ Inorden

Se procesa el primer hijo, luego la raíz y por último los restantes hijos

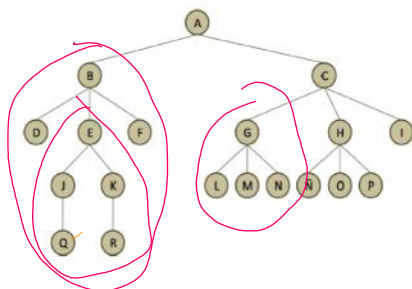
➤ Postorden

Se procesan primero los hijos y luego la raíz

➤ Por niveles

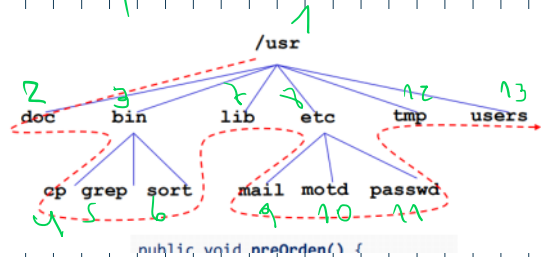
Se procesan los nodos teniendo en cuenta sus niveles, primero la raíz, luego los hijos, los hijos de éstos, etc.

Dado el siguiente árbol, escriba los recorridos preorden, inorden y postorden

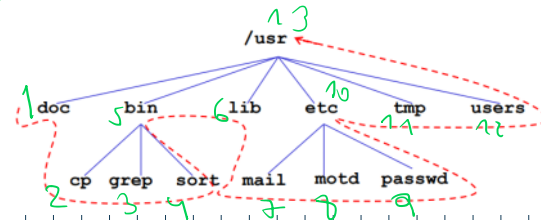


Preorden: A B D E J K R C G L M N H O P I
 Inorden: Q J R K E B F L M N G C H O P S A

Preorden



Postorden



Preorden: A B D E J K R C G L M N H O P I

Postorden: Q J R K E B F L M N G C H O P S A

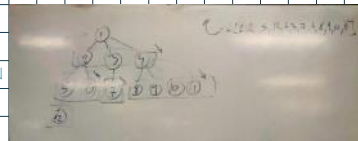
NOP.

INSDBJEKFA
 LAMNCLNHP

POST { D B J R K E F B
 L M N G N O P H I C A

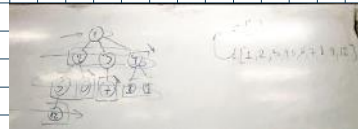
PRE { A B D E J R K F
 C G L M N H N O P I

```
package ayed;
public class ArbolGeneral<T> {
    public ListaEnlazadaGenerica<T> preOrden() {
        ListaEnlazadaGenerica<T> lis = new ListaEnlazadaGenerica<T>();
        this.preOrden(lis);
        return lis;
    }
    private void preOrden(ListaGenerica<T> l) {
        l.agregarFinal(this.getDato());
        ListaGenerica<ArbolGeneral<T>> lHijos = this.getHijos();
        lHijos.comenzar();
        while (!lHijos.fin()) {
            (lHijos.proximo()).preOrden(l);
        }
    }
}
```



```
public ListaGenerica<T> porNiveles(ArbolGeneral<T> arbol) {
    ListaGenerica<T> result = new ListaEnlazadaGenerica<T>();
    ColaGenerica<ArbolGeneral<T>> cola = new ColaGenerica<ArbolGeneral<T>>();
    ArbolGeneral<T> arbol_aux;
    cola.encolar(arbol);
    while (!cola.esVacia()) {
        arbol_aux = cola.desencolar();
        result.agregarFinal(arbol_aux.getDato());
        if (arbol_aux.tieneHijos()) {
            ListaGenerica<ArbolGeneral<T>> hijos = arbol_aux.getHijos();
            hijos.comenzar();
            while (!hijos.fin()) {
                cola.encolar(hijos.proximo());
            }
        }
    }
    return result;
}
```

x niveles



siempre q' tengo q' guardar lista, tengo q' hacer copia por f' no referencia a la misma y lo pierdo.

A la vez, al q' era q' guardar referencias por f' orden cuando vuelve para actualizar tengo que borrar el ult.

ej: 1 2 3 1 → 1 2 3
 2 3 2 → 1 2 3
 3 3 3 → 1 2 3
 borro el último.