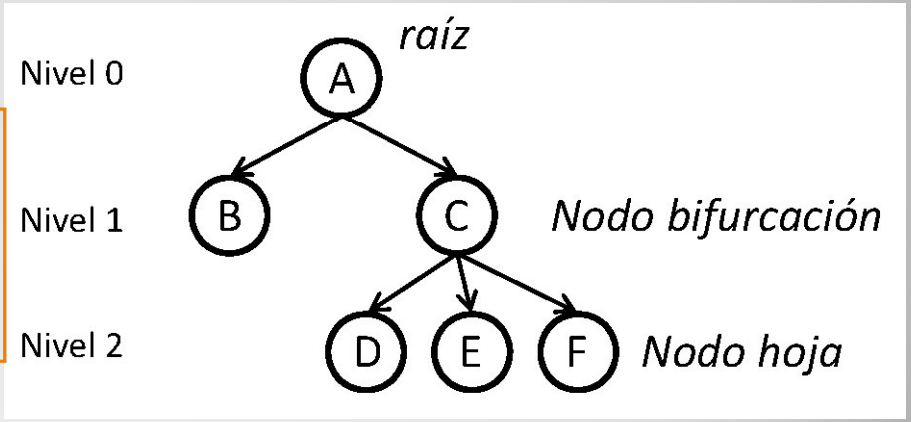
Curso C

Estructuras Dinámicas
Parte II

Arboles

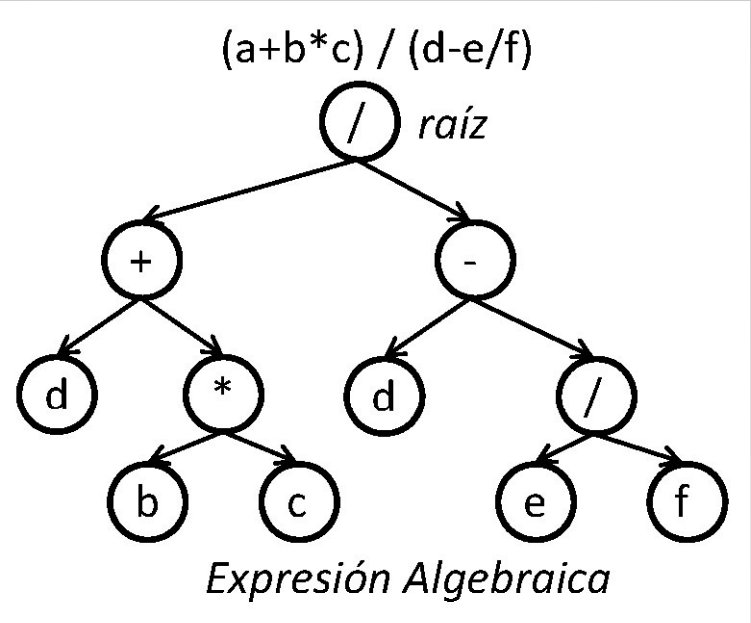
- Un árbol es una estructura NO lineal formada por un conjunto de nodos y un conjunto de ramas.
  - Nodo especial llamado raíz.
  - Nodo bifurcación, aquel del que sale alguna rama
  - Nodo terminal o hoja, aquel que no tiene ninguna rama.

• Profundidad o altura del árbol = nivel máximo
• Grado = numero de ramas de un nodo



Arboles binarios

- Consta de nodo raíz y dos subárboles binarios: subárbol izdo y subárbol dcho.
  - Grado máximo de todos los nodos es 2.

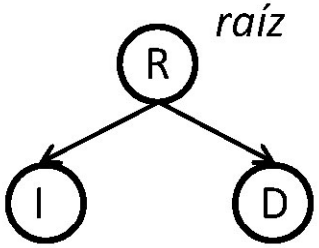


Ejemplo: Expresión algebraica

Arboles binarios

- Formas de recorrer arbol binario:
  - Preorden, Inorden, Postorden
  - Se trata de recorridos

RECURSIVOS



Preorden: R, I, D
Inorden: I, R, D
Postorden: I, D, R

```
typedef struct s{
   void *datos;
   struct s* izquierdo;
   struct s* derecho;
}Nodo;
```

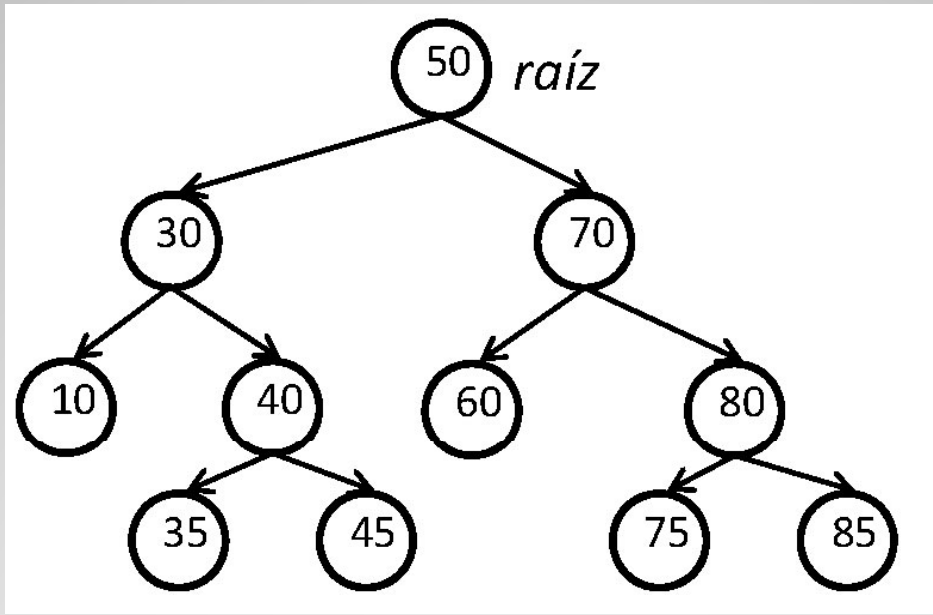
```
void preorden (Nodo* r){
 if(r != NULL){
    //operaciones con r (raiz)
    preorden(r->izquierdo);
    preorden(r->derecho);
 }}
```

```
void inorden (Nodo* r){
 if(r != NULL){
    inorden(r->izquierdo);
  //operaciones con r (raiz)
    inorden(r->derecho);
 }}
```

```
void postorden(Nodo* r){
 if(r != NULL){
    postorden(r->izquierdo);
    postorden(r->derecho);
//operaciones con r (raiz)
 }}
```

Arboles binarios de búsqueda

- Se trata de un árbol binario ordenado. Rama izda de nodo raíz menores que raíz, rama dcha de nodo raíz mayores que raíz.



Arboles binarios de búsqueda

- Interfaz:

```
typedef struct s{
   void *datos;
   struct s* izquierdo;
   struct s* derecho;
}Nodo;
```

Función	Interfaz
Nuevonodo	Nodo* Nuevonodo();
Insertar	int Insertar(void* datos, Nodo** raiz);
borrar	void *Borrar(void* datos, Nodo** raiz);
Buscar	void *Buscar(void* datos, Nodo* raiz);
Inorden	void Inorden(Nodo *raiz);
borraArbol	Recursiva, libera memoria de cada nodo, recorriendo en postorden. void borraArbol(Nodo** raiz);
Comparar	Funcion definida por usuario. Devuelve -1, 0 ó 1 si nodo1<nodo2, nodo1==nodo2 o nodo1>nodo2 int Comparar(void *datos1, void*datos2)
LiberarMemoria	Libera memoria reservada para void *datos de cada nodo. Usada en borraArbol

Arboles binarios perfectamente equilibrados

- Será perfectamente equilibrado si, para todo nodo, el número de nodos en el subárbol izdo y el número de nodos en el subárbol dcho difieren como mucho en una unidad.

- Forma de distribuir los datos para n nodos, siendo ni el numero de nodos a la izda y nd numero de nodos a la dcha.:
  - 1- Utilizar un nodo para la raíz
  - 2- generar subarbol izdo con ni=n/2 nodos
  - 3- generar subarbol dcha con nd=n-ni-1 nodos

Para poder controlar cuantos nodos llevamos, debemos incluir en los datos del nodo un int con el NumeroDeNodos.