



UNIVERSIDAD TECNOLÓGICA NACIONAL
INSTITUTO NACIONAL SUPERIOR DEL PROFESORADO TÉCNICO

ARBOLES

ESTRUCTURAS y BASE DE DATOS

Comisión 2-603

Profesores: Juan Capia

Alejandro Behringer

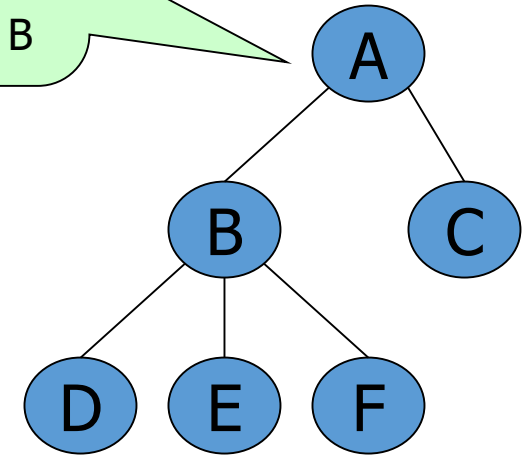
Introducción

- Las listas enlazadas son estructuras lineales

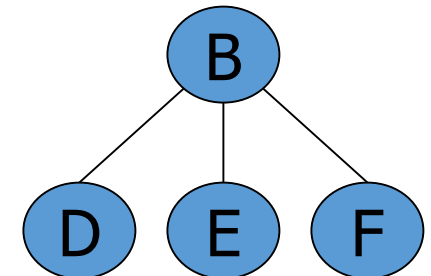
Son flexibles pero son secuenciales, un elemento detrás de otro.
- Los árboles
 - Junto con los grafos son estructuras de datos no lineales
 - Superan las desventajas de las listas .
 - Sus elementos se pueden recorrer de distintas formas, no necesariamente uno detrás de otro .
- Son muy útiles para la búsqueda y recuperación de información

Elementos principales

A es Padre
B y C hijos de A:
hermanos
B es Padre
D, E, F hijos de B



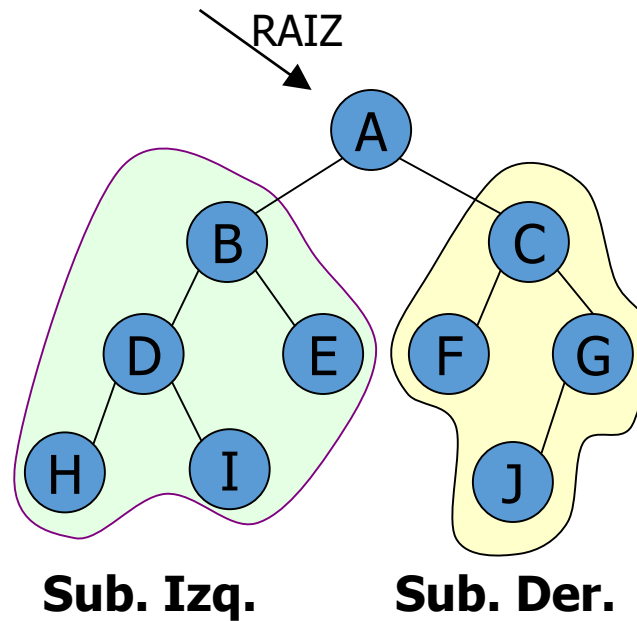
- Estructura que organiza sus elementos formando jerarquías: PADRES E HIJOS
- Los elementos de un árbol se llaman nodos
- Si un nodo p tiene un enlace con un nodo m,
- Todos los nodos tienen al menos un padre, menos la raíz: A
 - ❑ Si no tienen hijos se llaman **hoja**: D, E, F y C
 - ❑ Los hijos de un mismo padre se llaman: hermanos
- Un subárbol de un árbol
 - ❑ Es cualquier nodo del árbol junto con todos sus descendientes



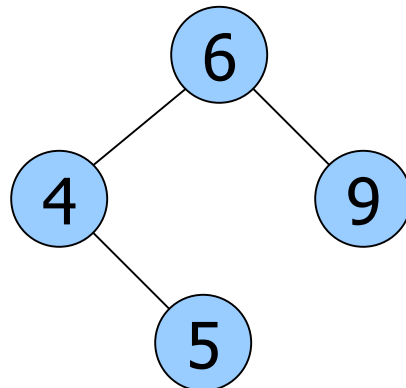
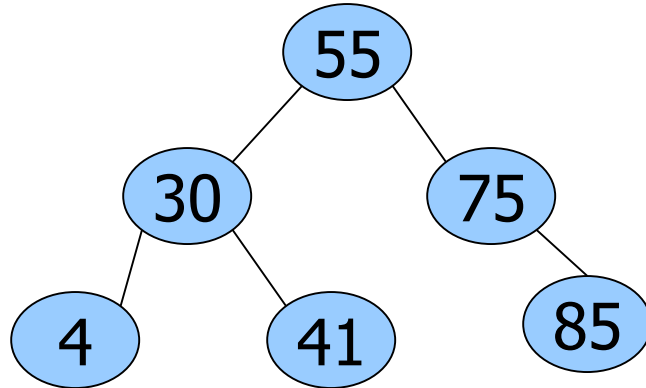
Elementos principales

- **Altura de un nodo** en un árbol es el número de nodos del camino más largo de ese nodo a una hoja.
- **Altura del árbol** es la altura de la raíz, o 0 si el árbol es vacío.
- **Grado de un nodo** es el número de hijos de dicho nodo.
- **Grado de un árbol** es el mayor grado de los nodos que contiene.
- El **nivel de un nodo** se asigna en función al criterio siguiente:
La raíz tiene nivel 1. Si un nodo tiene nivel N , sus hijos tendrán nivel $N+1$.
- Nivel de un árbol es igual a la altura de su raíz, o a 0, si el árbol es vacío.

Arboles binarios



Árbol binario de búsqueda



- Los elementos en un árbol
 - Hasta ahora no han guardado un orden.
 - No sirven para buscar elementos.
- Los arboles de búsqueda
 - Permiten ejecutar en ellos búsqueda binaria
 - Dado un nodo:
 - Todos los nodos del sub. Izq. Tienen una clave menor que la clave de la raíz.
 - Todos los nodos del sub. Der. Tienen una clave mayor que la clave de la raíz.

Estructura de un nodo

- Como el nodo ahora tiene un campo dato y podrá tener o no nodos hijos .

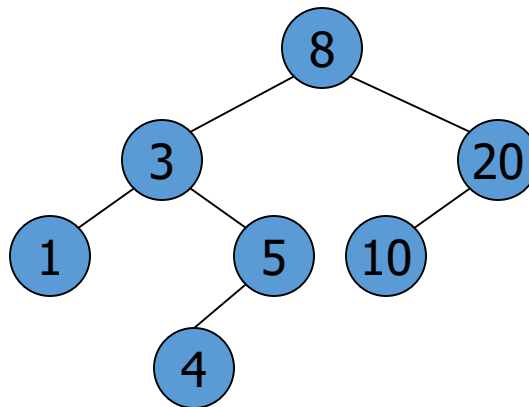
```
typedef struct NodoArbolBB
{
    int dato,
    NodoArbolBB * izq,
    NodoArbolBB * der,
}
```

RECORRIDOS DE UN A.B.

- Recorrer es
 - Visitar todos los elementos de una estructura
- Como recorrer un árbol
 - Cuantos caminos, hay para elegir?
 - Existe tres recorridos típicos
 - Nombrados de acuerdo a la posición de la raíz
 - Preorden: raíz - subárbol izq. - subárbol der. R - I - D
 - Inorden : subárbol izq. - raíz - subárbol der. I - R - D
 - Postorden : subarbol izq. - subarbol der. –raíz I - D - R

Creación de un ABB

- Un árbol de búsqueda debe mantener
 - A la derecha mayor a la raíz
 - A la izq. menor a la raíz
- Ejemplo:
 - Construya árbol con los siguientes elementos:
 - 8, 3, 1, 20, 10, 5, 4



Ejercicio en clase

- Construya el árbol para almacenar:

60	28	40	80	74	59	52	53	63	67	98	55
----	----	----	----	----	----	----	----	----	----	----	----

- Luego realizar las siguientes operaciones :

Eliminar el nodo 55
Eliminar el nodo 63
Eliminar el nodo 80
Eliminar el nodo 59

Agregar el nodo 45
Agregar el nodo 48
Agregar el nodo 43
Agregar el nodo 47
Agregar el nodo 49
Eliminar el nodo 52
Eliminar el nodo 45

Construir un árbol binario de búsqueda ,inicialmente vacio con los siguientes valores :

a)

37,38,2,1,20,3,17,26,27,9,12,97,46,62,97,76,51,47,72,79,65,70,73,74,8
9,81,96,83

Luego :

Eliminar 76 ,79,72,62

b)

25,3,86,2,5,92,38,19,27,64,10,31,40,71,8,14,39,62,66,81,45,65,72,78,4
5,43,42

Eliminar cualquier nodo a elección que considere los tres casos posibles y además determinar la altura del árbol ,cuantos niveles tiene ,cuantos nodos hoja tiene ,decir si el árbol esta balanceado o no ,completo o no .