



UNIVERSIDAD
NACIONAL DE
HURLINGHAM

Estructuras de Datos

Profesor

Sergio Gonzalez

Unidad 7: Arboles

Profesor
Sergio Gonzalez

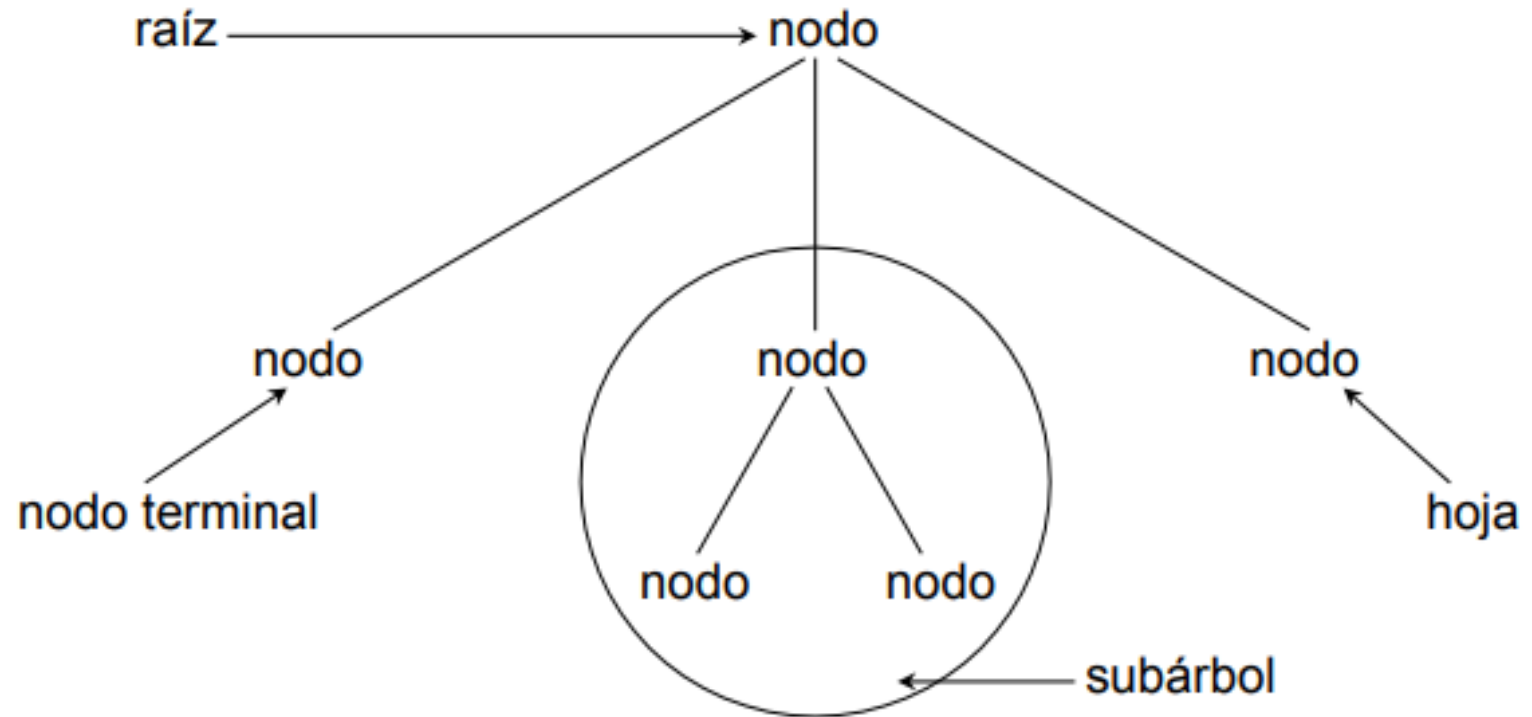
Estructuras dinámicas no lineales

- Cada elemento relacionado con mas de uno por delante o por detrás

Árboles

- Estructura jerárquica
- Definición recursiva:
 - Cada nodo posee N nodos siguientes
 - Cada uno de ellos es un árbol (subárbol)

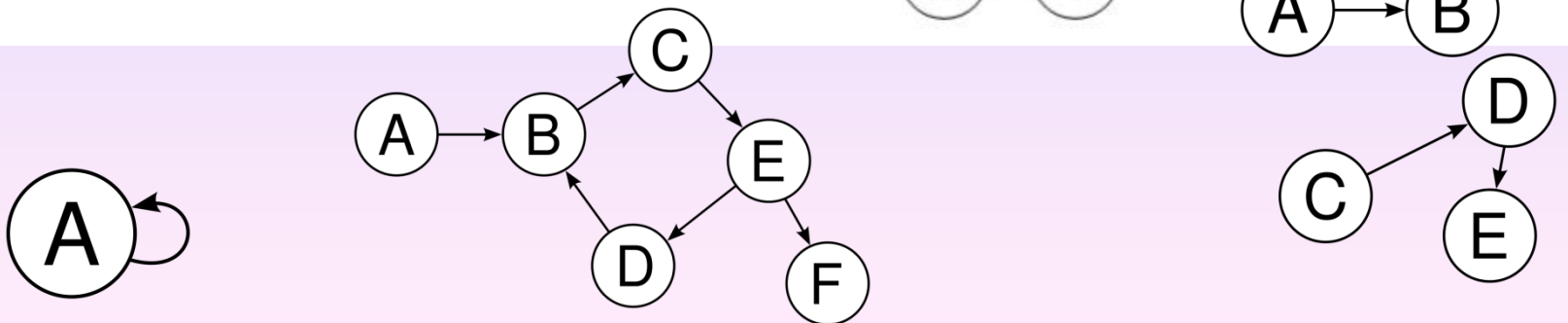
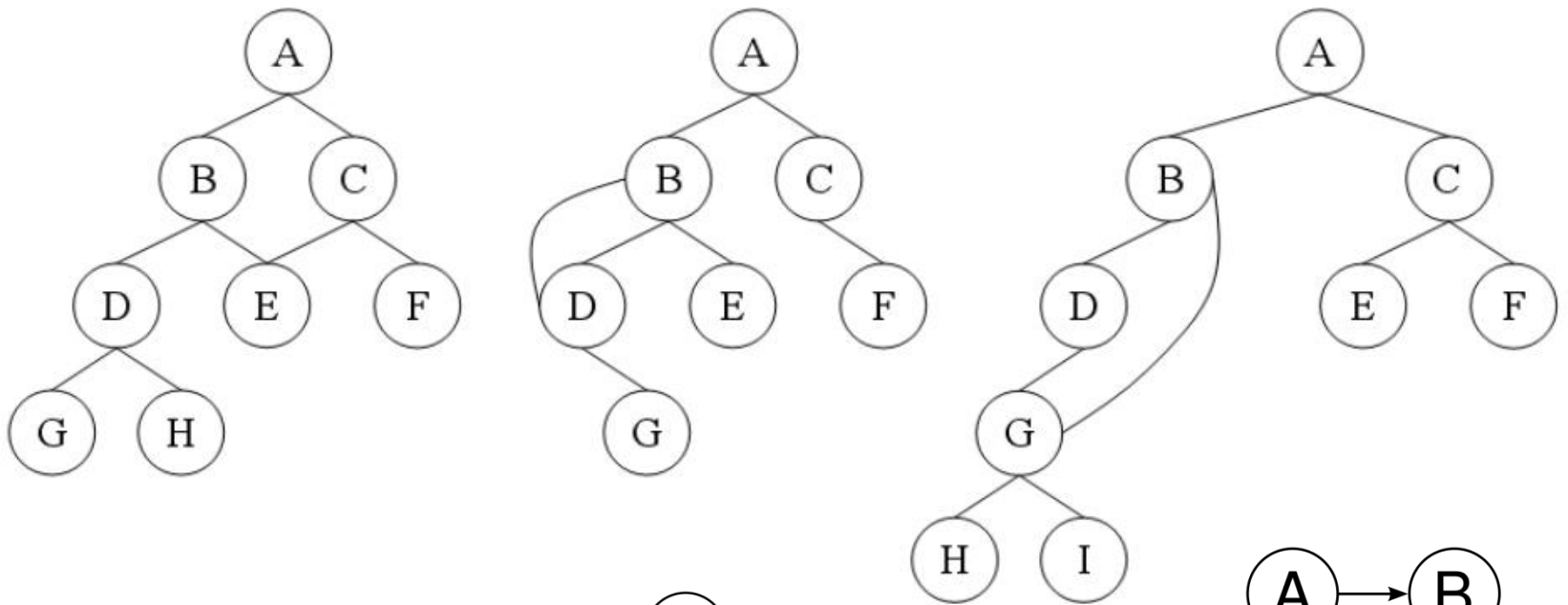
Árboles



Árboles

- Cada nodo tiene un único padre y puede tener mas de un hijo
 - Prohibido padres múltiples
 - No hay ciclos
- Desde la raíz puedo llegar a cualquier nodo interno

Árboles



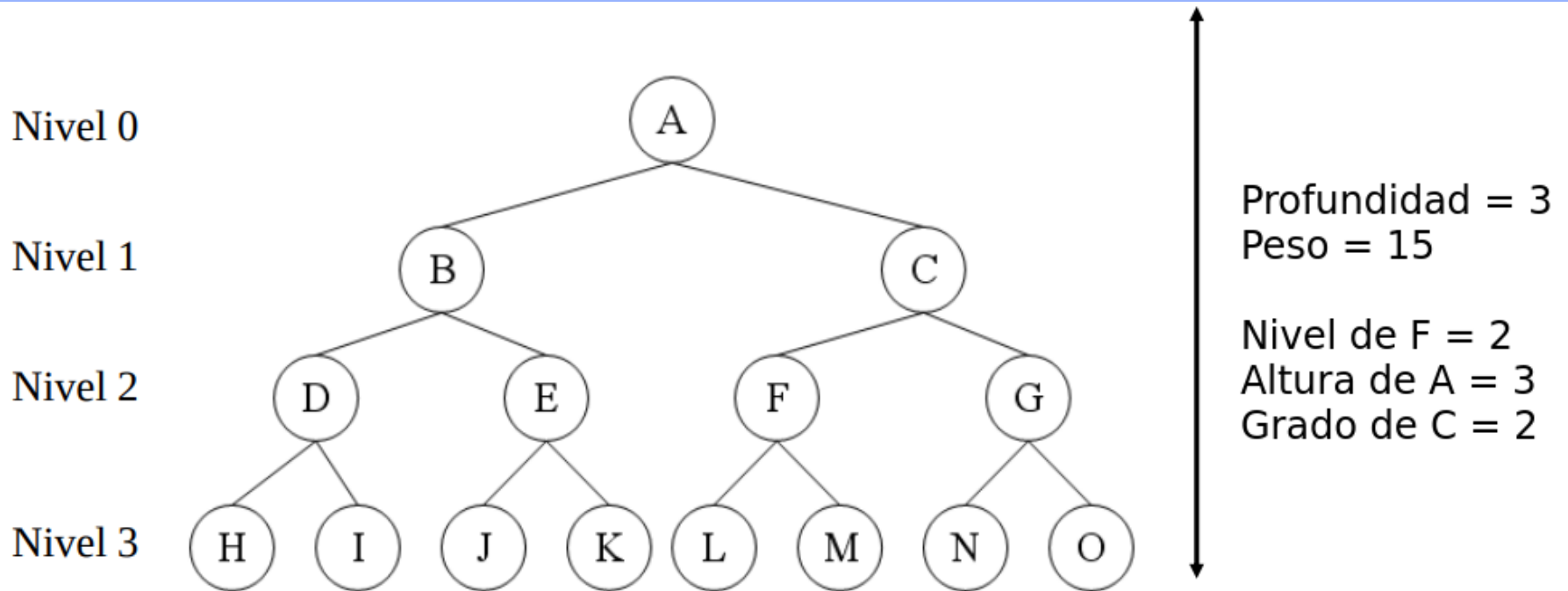
Árboles

- Algunas definiciones:
 - Trayectoria entre 2 nodos : Secuencia de nodos entre ellos.
 - Largo de trayectoria: Número de enlaces
 - Grado de un nodo: Cantidad de subárboles hijos
 - Peso de un árbol: Cantidad total de nodos
 - Altura de un nodo: Largo de trayectoria a la hoja mas lejana
 - Profundidad de un nodo: Largo de trayectoria desde la raíz

Árboles

- Algunas definiciones:
 - Profundidad de árbol: Prof. de hoja mas profunda
 - Nodos a una misma profundidad -> Están al mismo nivel
- Altura raíz = Profundidad de árbol
- Grado de hojas = 0

Árboles



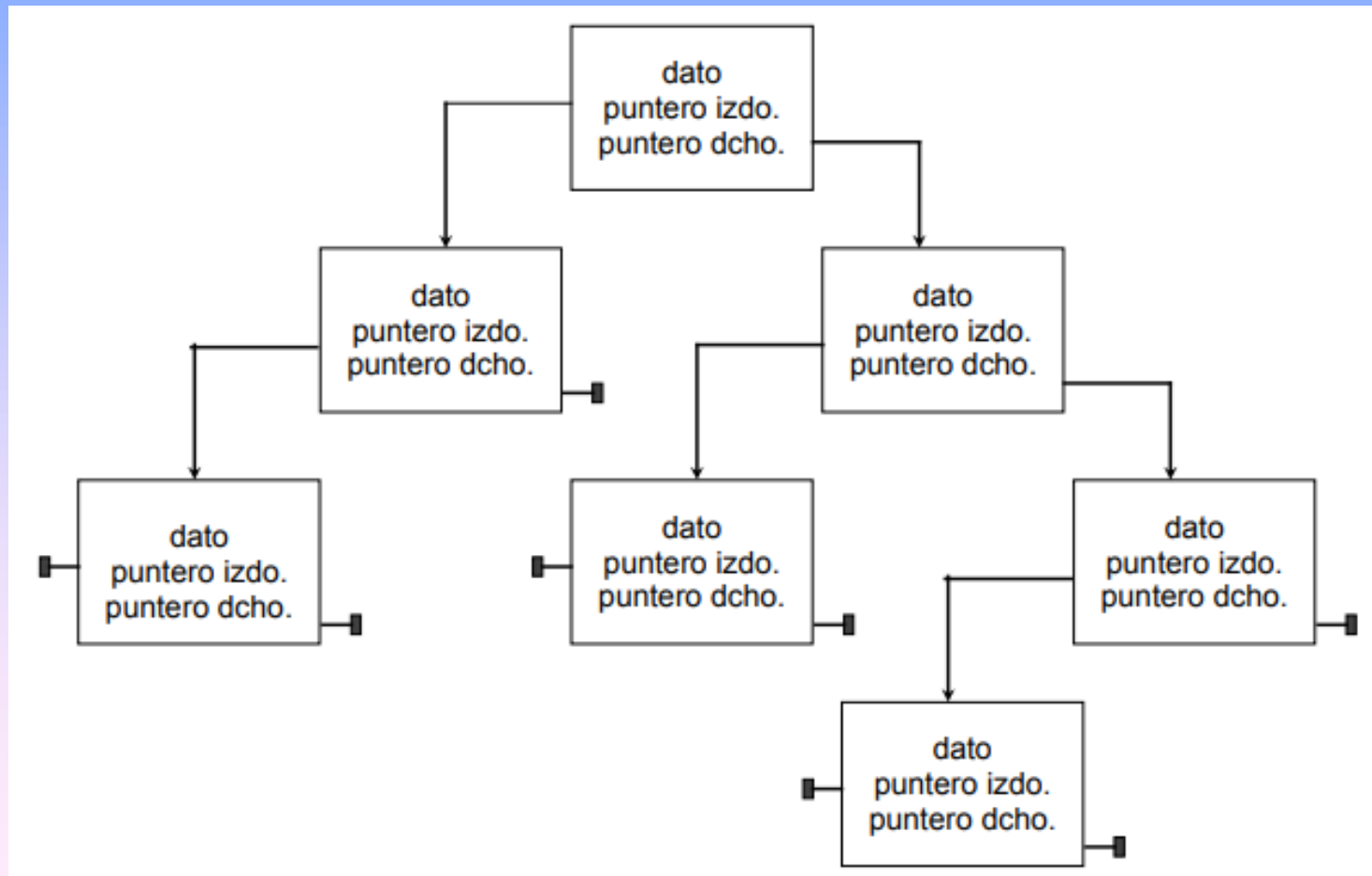
Árboles binarios

- Cada nodo posee a lo sumo dos subárboles binarios hijos:
 - Subarbol derecho
 - Subarbol izquierdo

Árboles binarios

- Cada nodo posee tres campos:
 - Dato
 - Subárbol derecho (puede ser null)
 - Subárbol izquierdo (puede ser null)

Árboles binarios



Árboles binarios

- Recorridos:
 - Acceder a todos los elementos de forma sistemática:
 - Visitar nodo raíz
 - Recorrer el subárbol derecho
 - Recorrer el subárbol izquierdo

Árboles binarios

- Recorridos:
 - Pre – orden
 - Post – orden
 - In – orden
- Según el momento en el que se visita el nodo raíz

Árboles binarios

- Recorrido pre – orden:
 - Visitar raíz
 - Recorrer subárbol izquierdo en pre – orden
 - Recorrer subárbol derecho en pre – orden

Árboles binarios

- Recorrido in – orden:
 - Recorrer subárbol izquierdo en in – orden
 - Visitar raíz
 - Recorrer subárbol derecho en in – orden

Árboles binarios

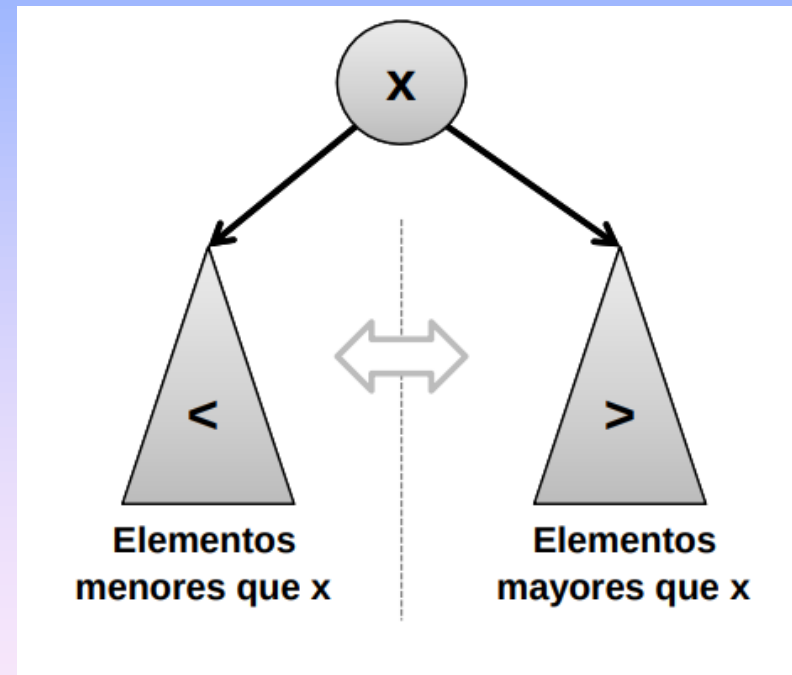
- Recorrido post – orden:
 - Recorrer subárbol izquierdo en post – orden
 - Recorrer subárbol derecho en post – orden
 - Visitar raíz

Tipos de Árboles

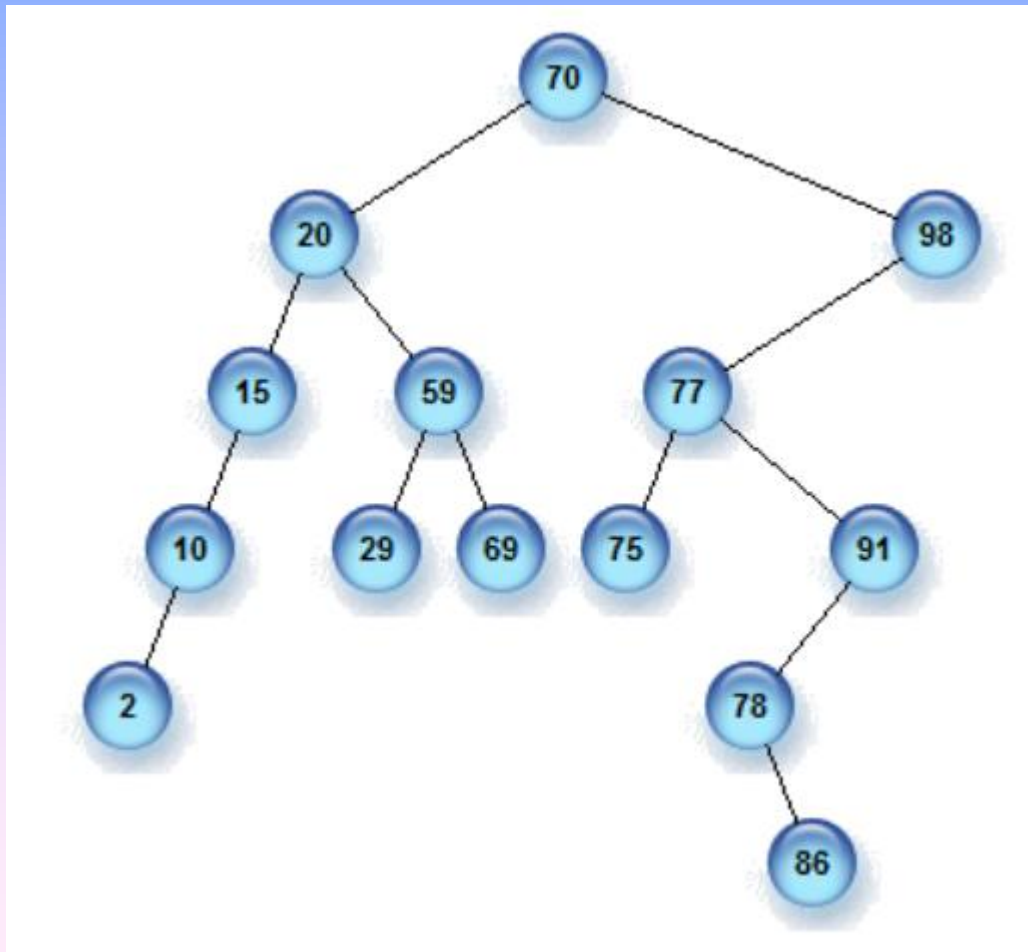
- Binario de búsqueda
- Balanceados
- AVL
- N-ario

Árboles binarios de búsqueda

- Datos en los nodos del subárbol izquierdo menores a la clave de la raíz
- Datos en los nodos del subárbol derecho mayores a la clave de la raíz
- No se aceptan claves duplicadas



Árboles binarios de búsqueda



Árboles binarios de búsqueda

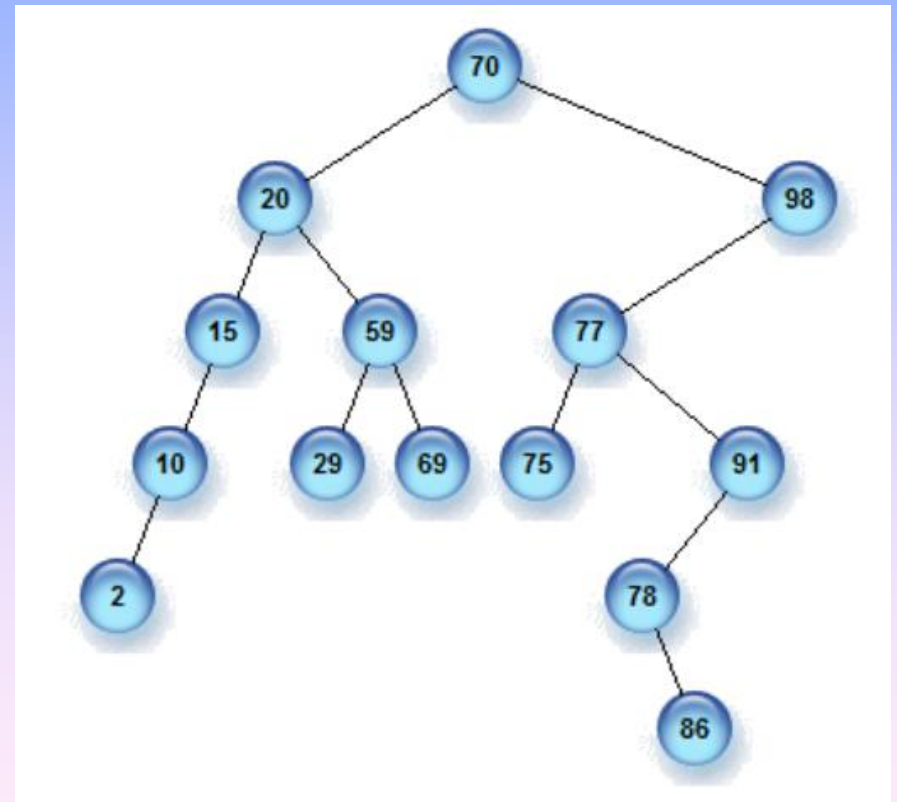
- Recorrido in – orden: Recorre los elementos de menor a mayor

Árboles binarios de búsqueda

- Búsqueda: Se compara con la raíz y se desciende por el subárbol correspondiente
- Inserción: Se busca en el árbol y se inserta como hoja

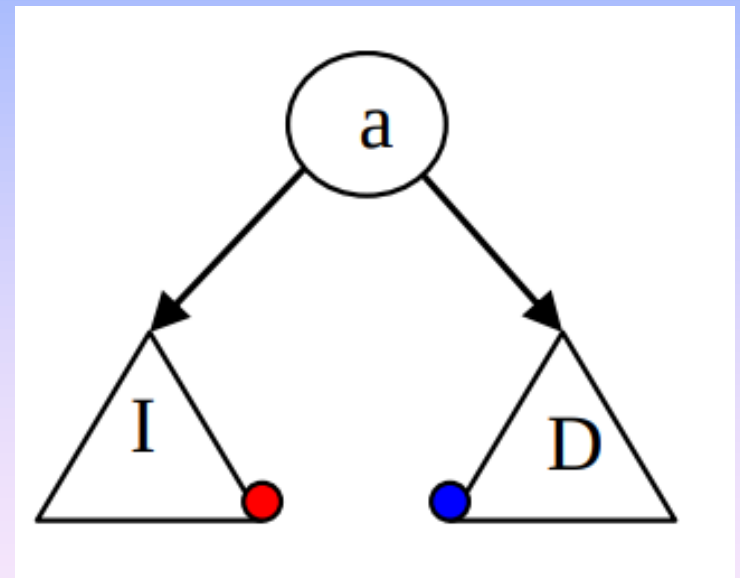
Árboles binarios de búsqueda

- Mínimo: Primer nodo sin hijo izquierdo
- Máximo: Primer nodo sin hijo derecho



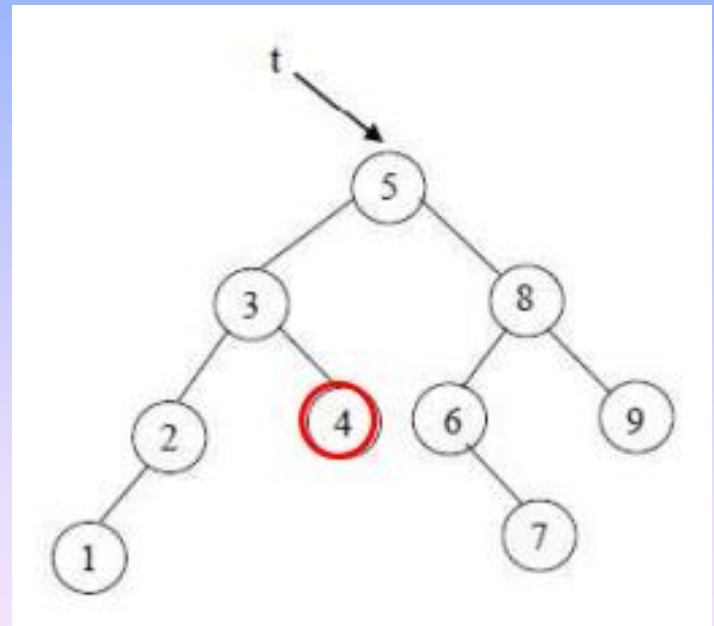
Árboles binarios de búsqueda

- Predecesor de un nodo:
Máximo del subárbol izquierdo
- Sucesor de un nodo:
Mínimo del subárbol derecho



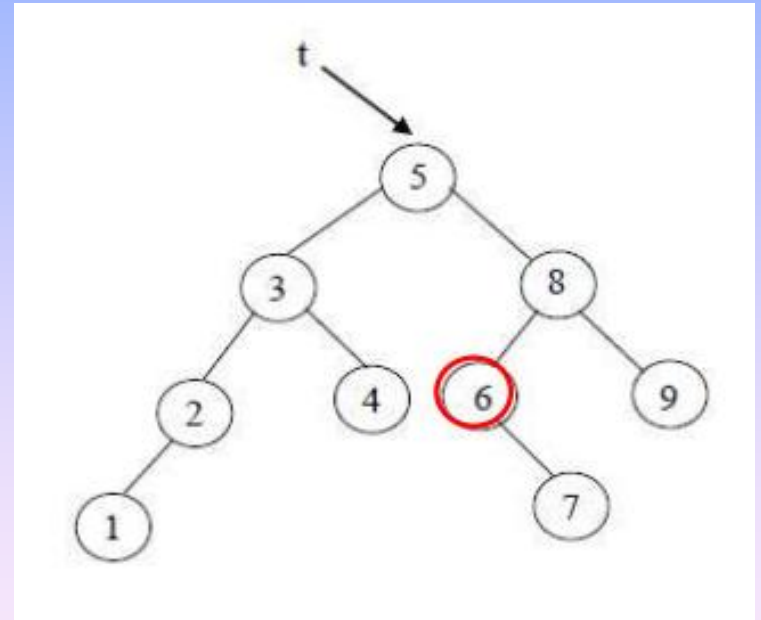
Árboles binarios de búsqueda

- Predecesor de un nodo:
Máximo del subárbol
izquierdo



Árboles binarios de búsqueda

- Sucesor de un nodo:
Mínimo del subárbol
derecho



Implementar TDA ABB con operaciones:

- Eliminar árbol / Vaciar
- EstaVacio
- Insertar
- Mostrar In-orden
- Mostrar Post-orden
- Mostrar Pre-orden
- Buscar elemento
- Calcular máximo
- Calcular mínimo
- Predecesor de un nodo (NodoArbol)
- Sucesor de un nodo (NodoArbol)
- Eliminar elemento
- Recorrer mostrando elemento y nivel
- Calcular grado de un nodo (NodoArbol)
- Calcular peso del árbol
- Calcular altura de un nodo (NodoArbol)
- Calcular profundidad del árbol
- tieneIzq, tieneDer, esHoja (NodoArbol)