

Estructuras de Datos

Profesor
Sergio Gonzalez

Unidad 7: Arboles

Profesor
Sergio Gonzalez

Estructuras dinámicas no lineales

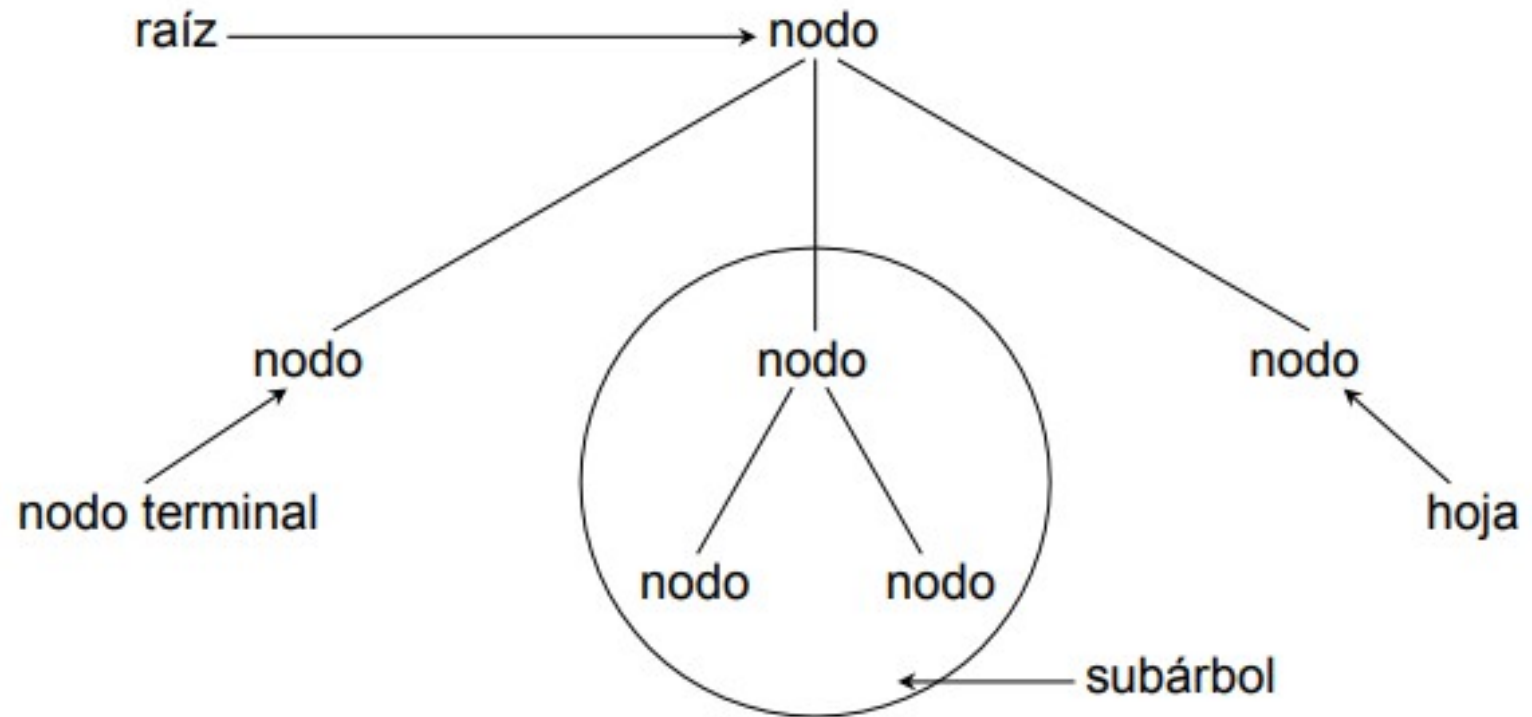
- Cada elemento relacionado con mas de uno por delante o por detrás

Árboles

- Estructura jerárquica
- Definición recursiva:
 - Cada nodo posee N nodos siguientes
 - Cada uno de ellos es un árbol (subárbol)
 - Nodo Raíz (Inicio)
 - Padre
 - Hijo
 - Nodo interno
 - Hoja (Nodos terminales)



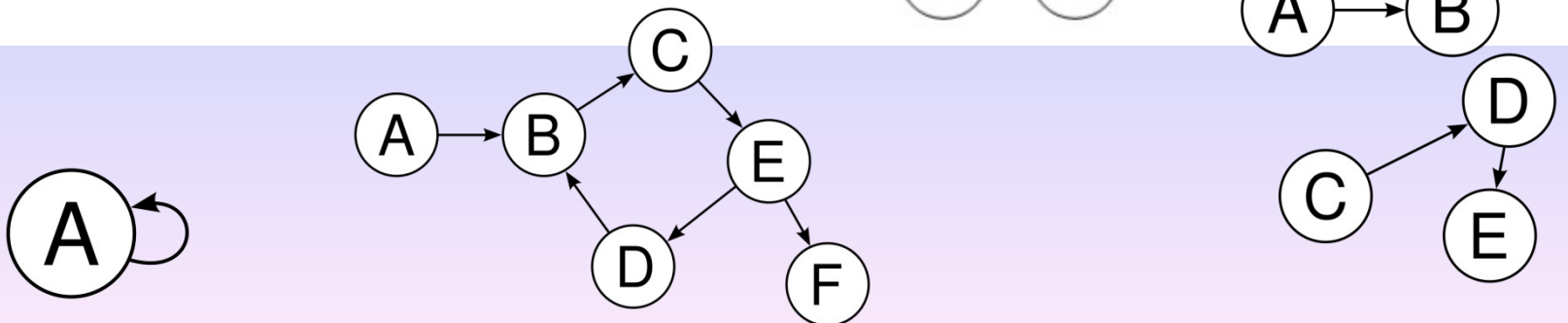
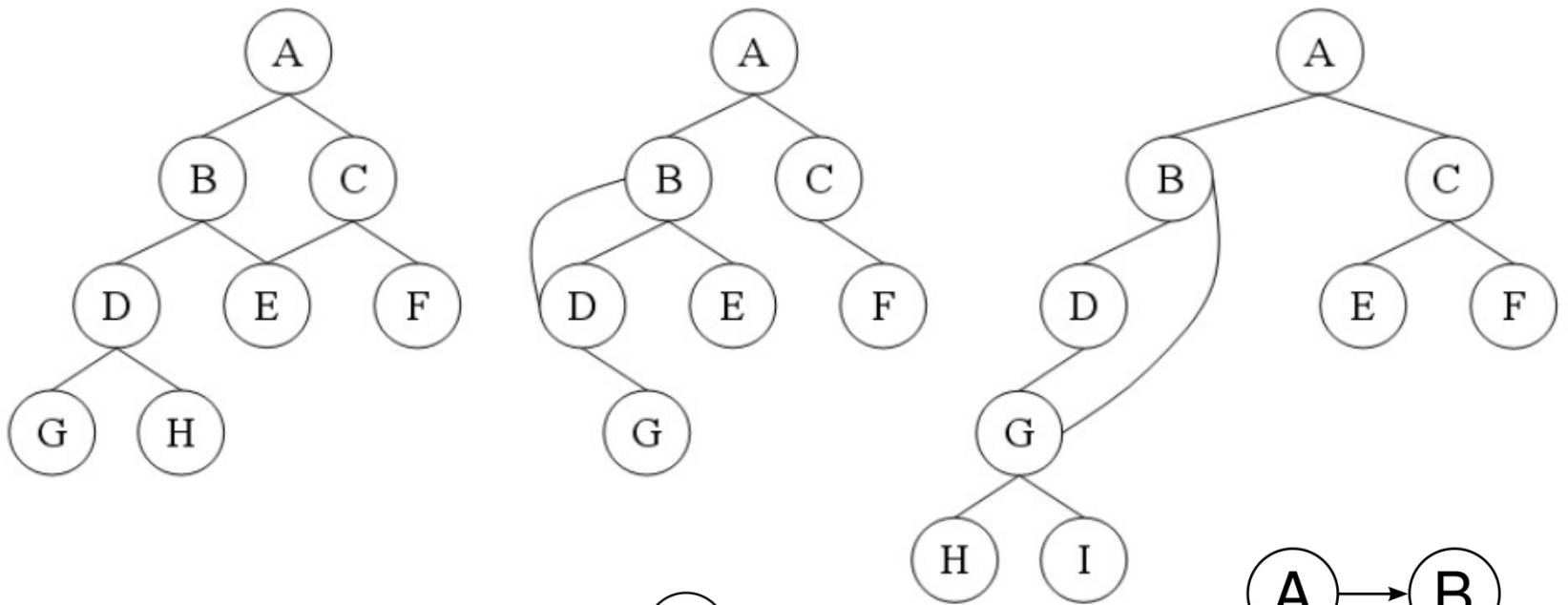
Árboles



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



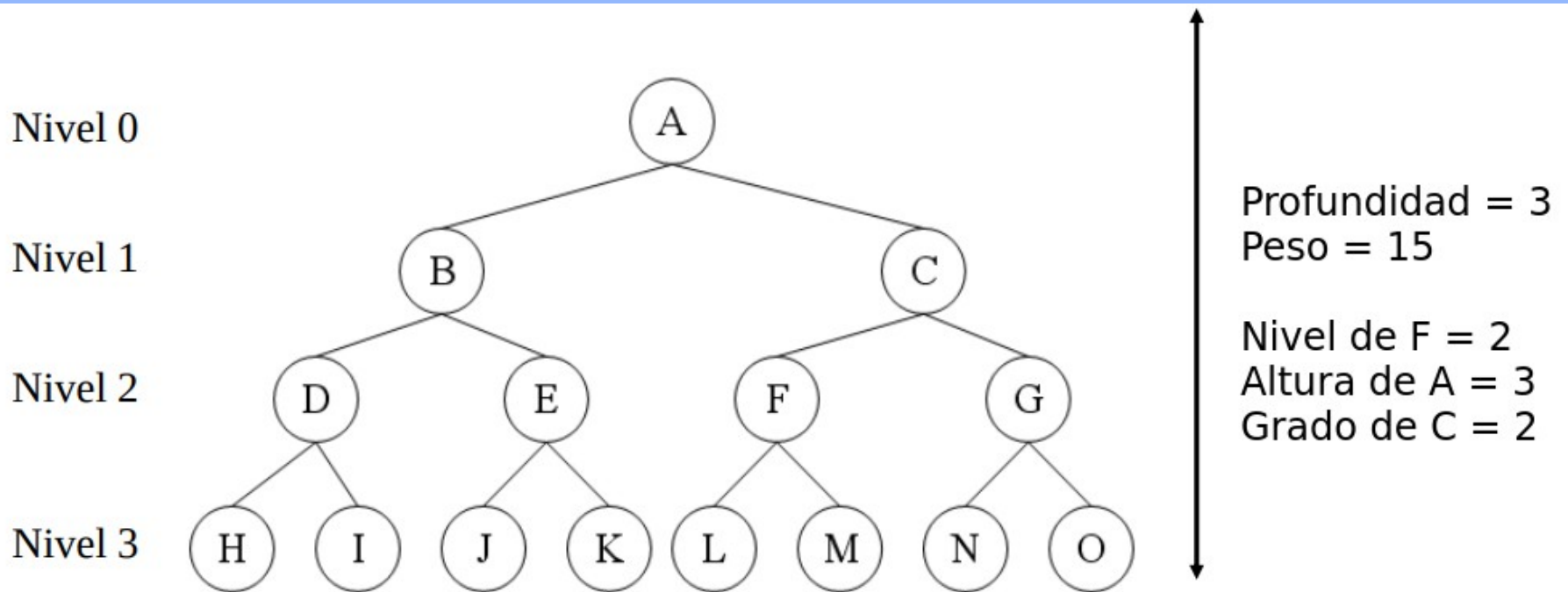
Árboles

- Algunas definiciones:
 - Trayectoria entre los nodos n y m : 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
 - Profundidad de árbol: Prof. de hoja mas profunda
 - Nodos a una misma profundidad -> Están al mismo nivel

Árboles

- Algunas definiciones:
 - Altura raíz = Profundidad de árbol
 - Grado de hojas = 0

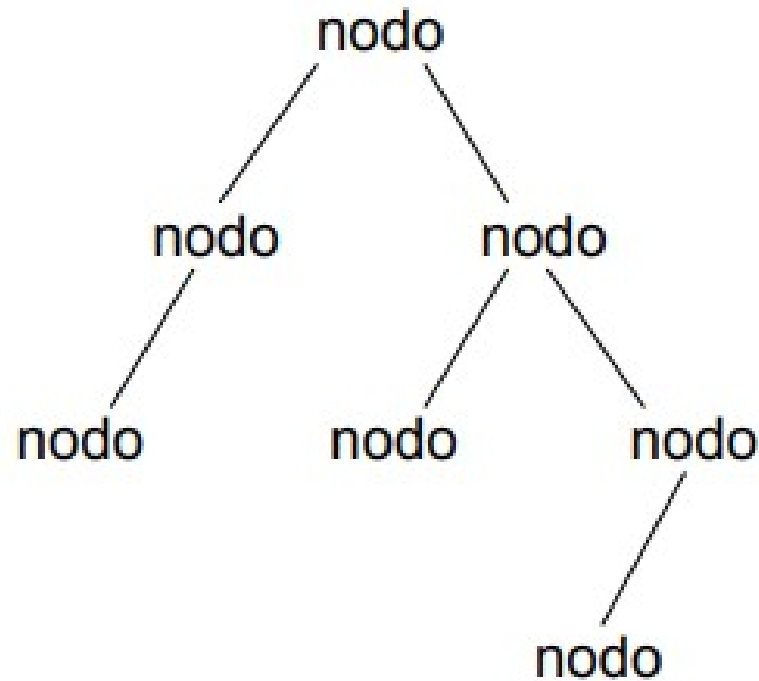
Árboles



Árboles binarios

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

Árboles binarios

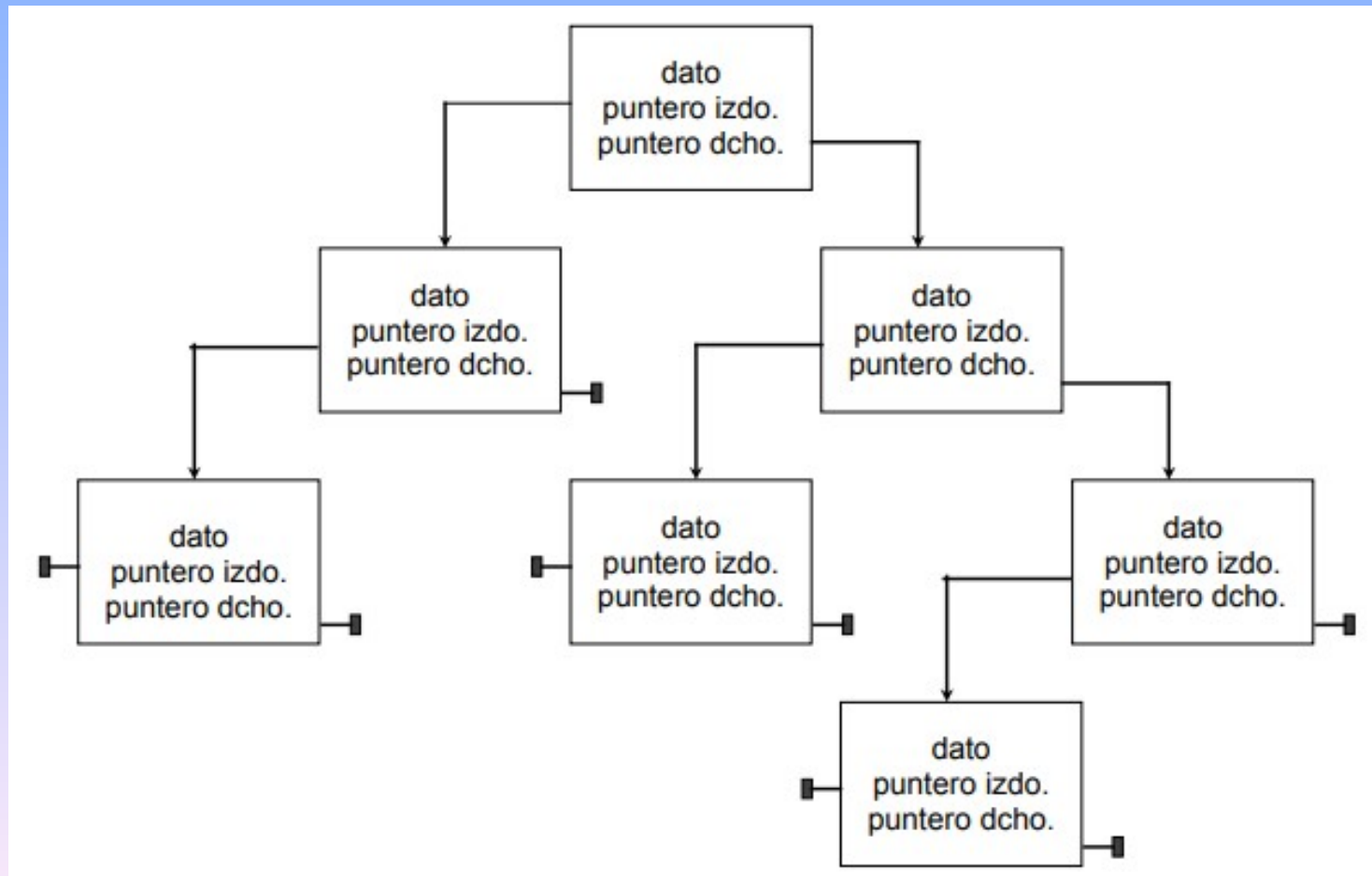


Árboles binarios

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



Árboles binarios



Árboles binarios

- Implementemos el TDA ArbolBinario en Python en el pizarrón

Árboles binarios: Operaciones

- Recorrer
- CalcularTrayectoria
- CalcularPeso
- CalcularProfundidad
- InsertarElemento (con algún criterio / condición)
- EliminarElemento
- BuscarElemento
- ModificarElemento
- Minimo / Maximo
- Sucesor / Predecesor

Árboles binarios

- Recorridos:
 - Acceder a 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

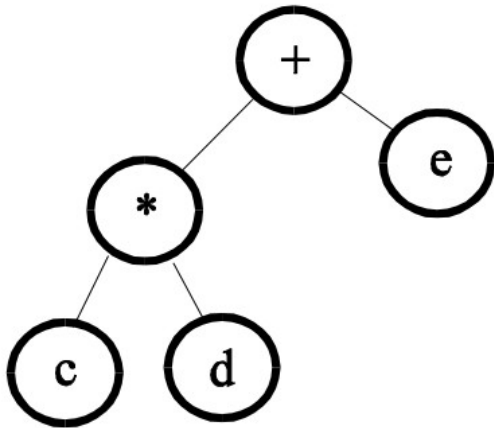
Árboles binarios

- Que tipo de algoritmo son los recorridos????

Árboles binarios

- Implementemos los recorridos en el pizarrón imprimiendo el dato al visitar un nodo

Árboles binarios



Pre-orden
In-orden
Post-orden

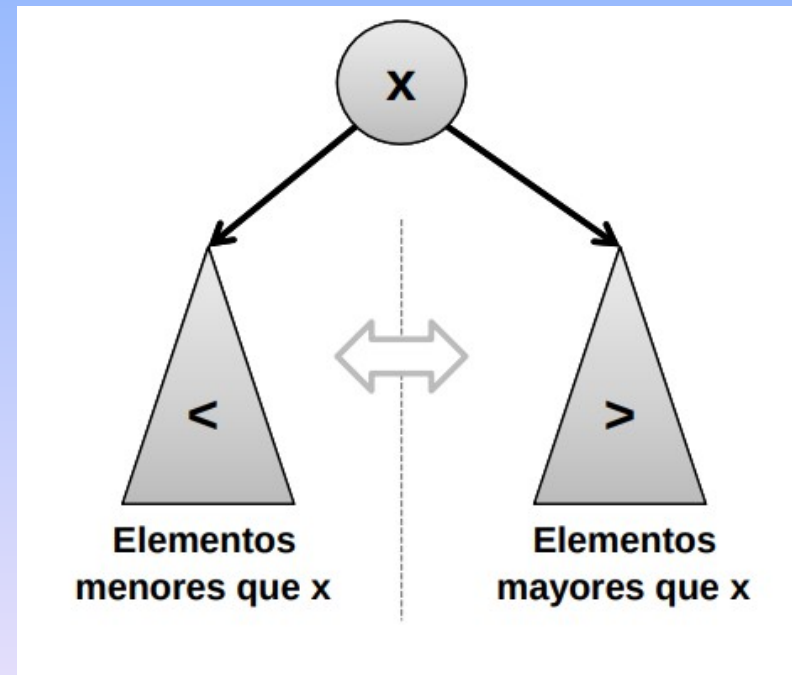
+ * cde
c * d + e
cd * e +

Tipos de Árboles

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

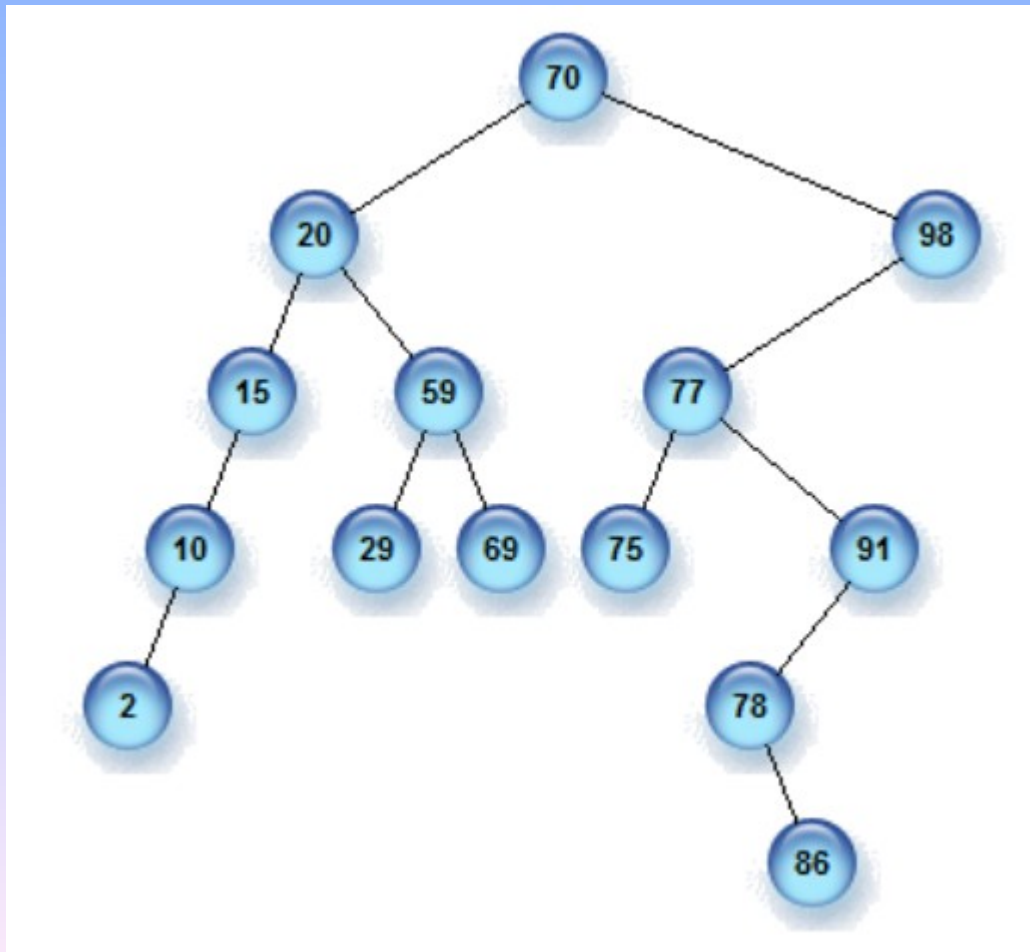
Árboles binarios de búsqueda

- Claves de los nodos del subárbol izquierdo menores a la clave de la raíz
- Claves de 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

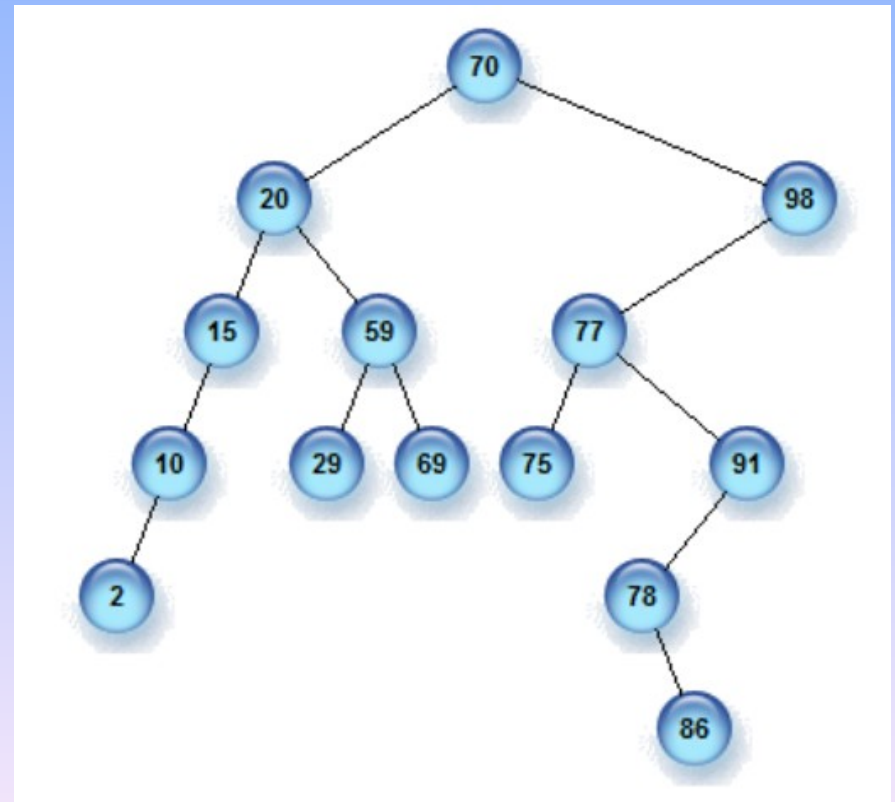
- Recorrido in – orden: Recorre los elementos de menor a mayor

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

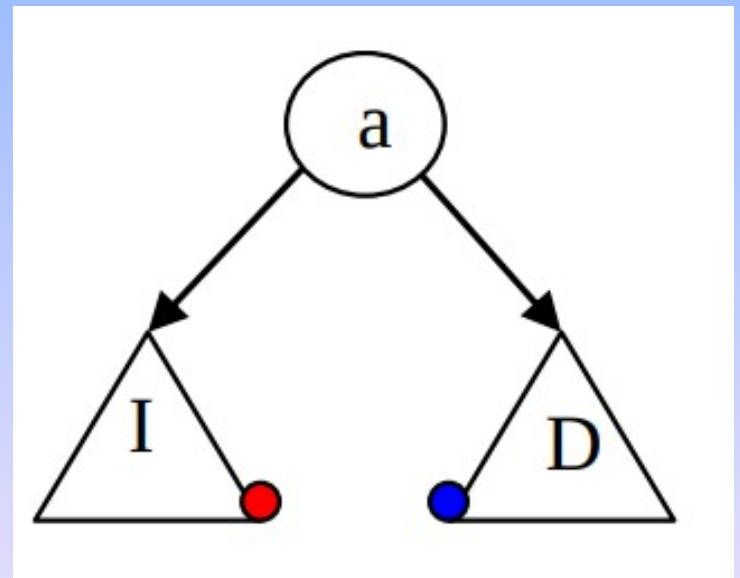
Árboles binarios de búsqueda

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



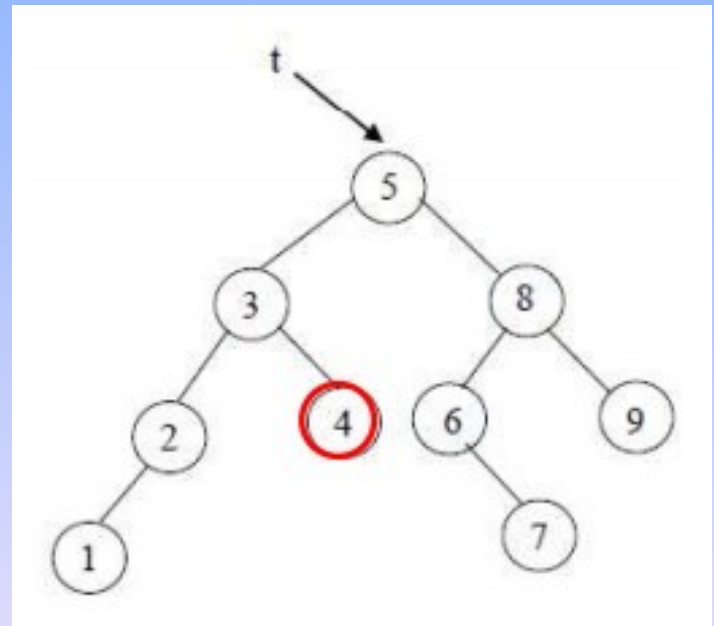
Árboles binarios de búsqueda

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



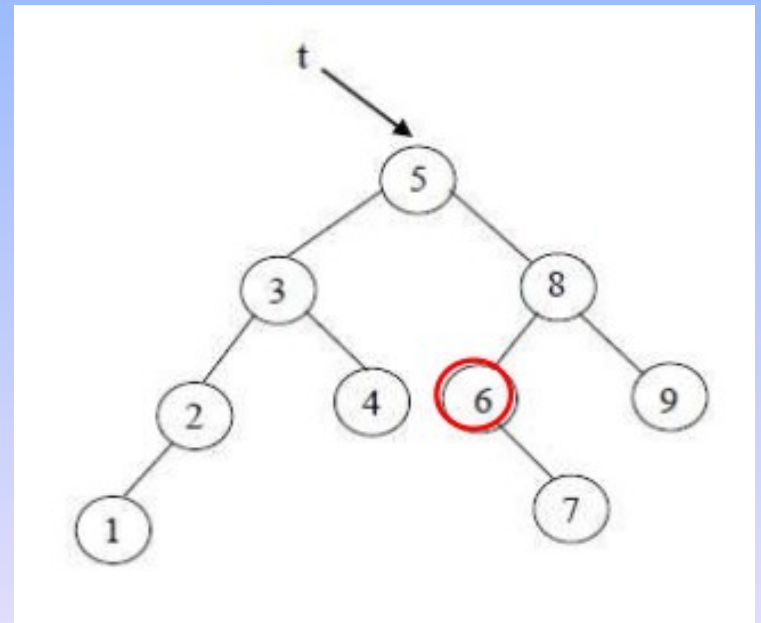
Árboles binarios de búsqueda

- Predecesor de un nodo:
Máximo del subárbol
izquierdo



Árboles binarios de búsqueda

- Sucesor de un nodo:
Mínimo del subárbol
derecho

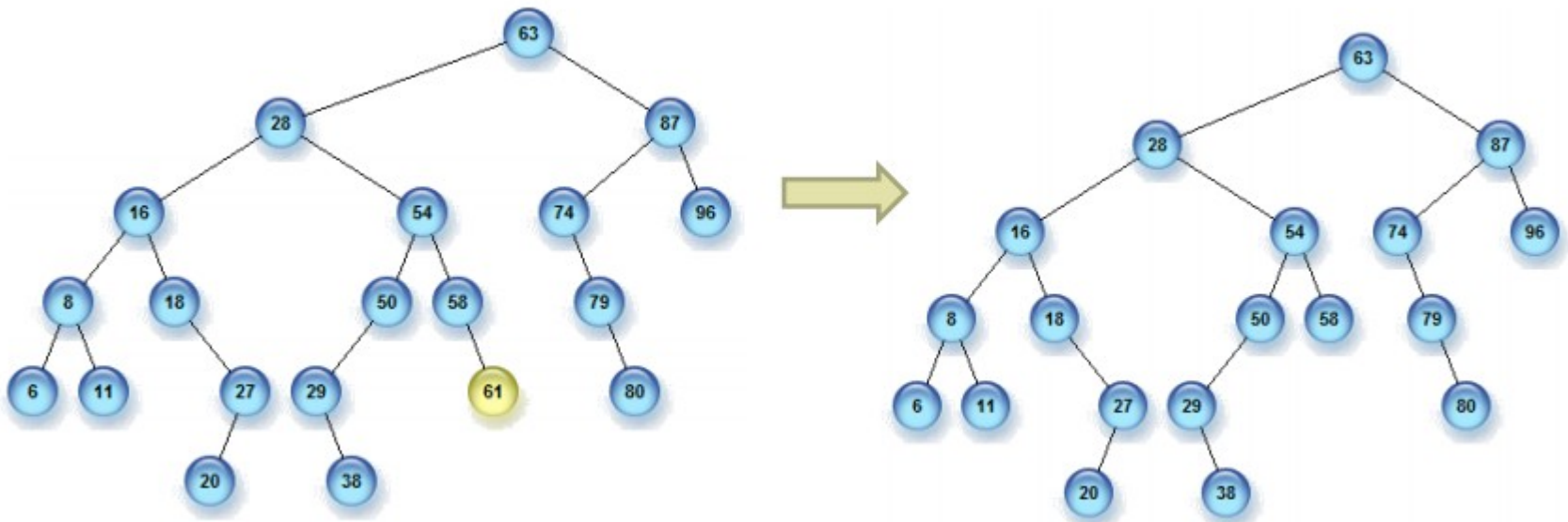


Árboles binarios de búsqueda

- Borrado:
 - Si tiene 0 o 1 hijo, se mueven los enlaces
 - Si tiene dos hijos, se intercambia con el máximo del subárbol izquierdo (predecesor).

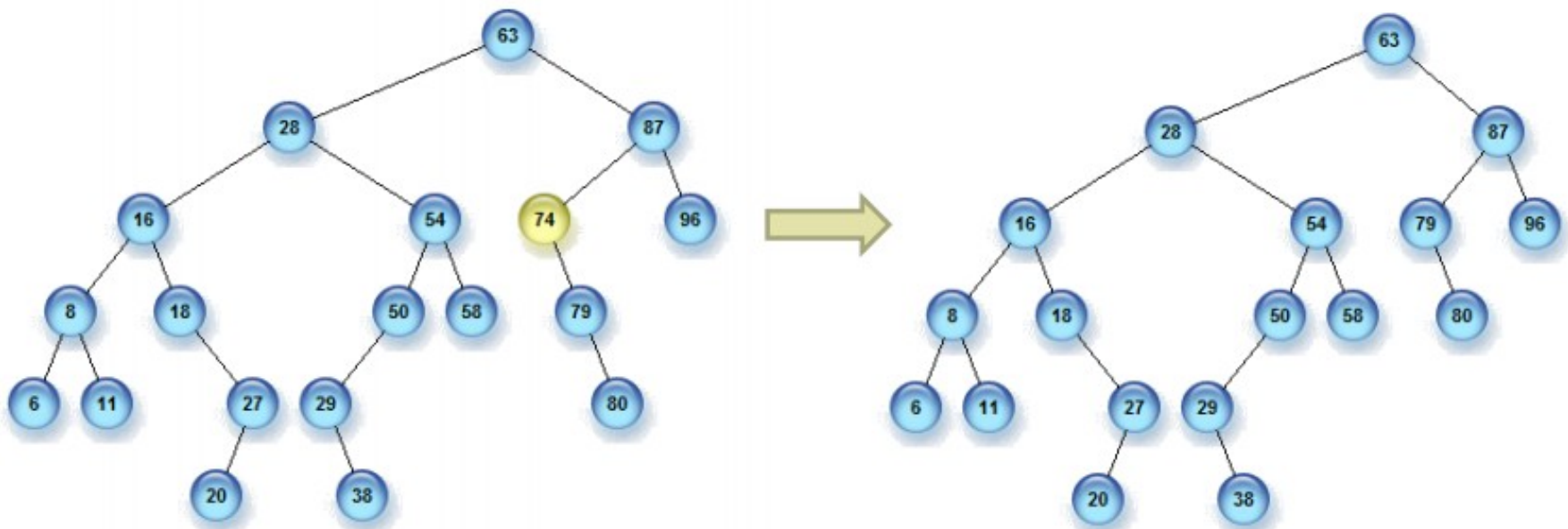
Árboles binarios de búsqueda

- Borrado con 0 hijos (hojas)



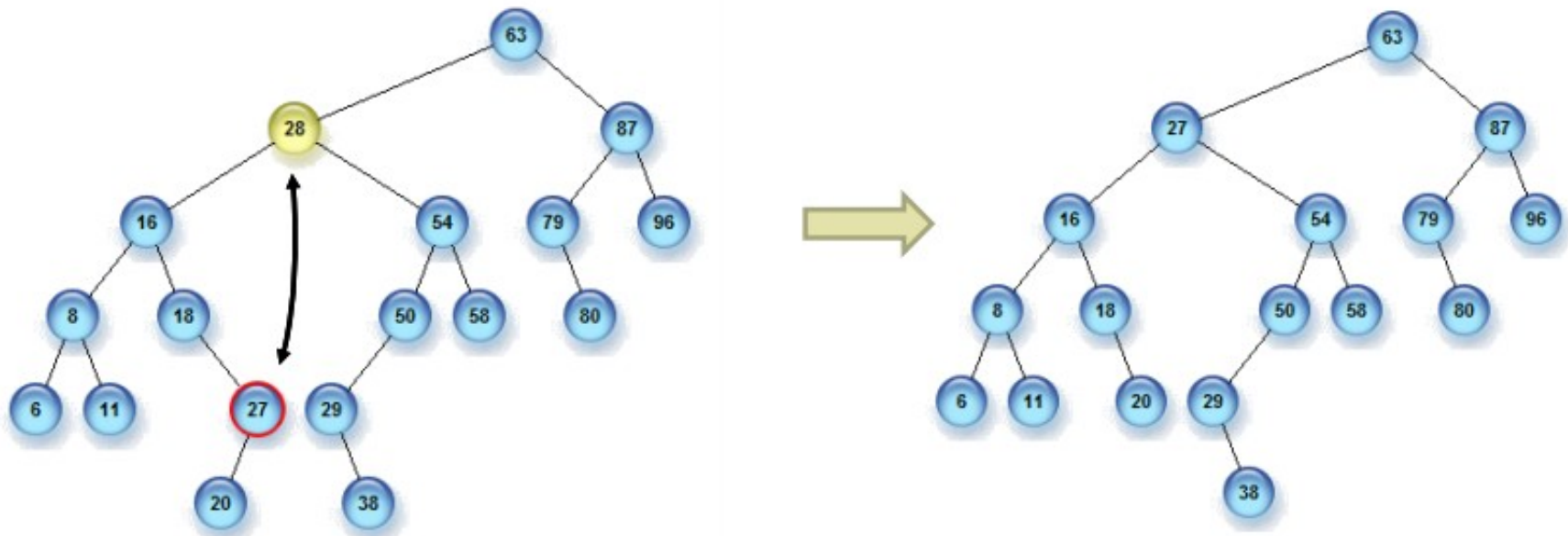
Árboles binarios de búsqueda

- Borrado con 1 hijo



Árboles binarios de búsqueda

- Borrado con 2 hijos



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

Balanceo en árboles

- Árbol balanceado: Altura logarítmica $\lfloor \log_2 N \rfloor$
- El balance del ABB depende del orden de inserción
- Elementos en orden: Peor caso (Árbol lista)
- Los ABB naturalmente no son balanceados
- Si se insertan elementos al azar, en promedio, son balanceados

Árboles Balanceados

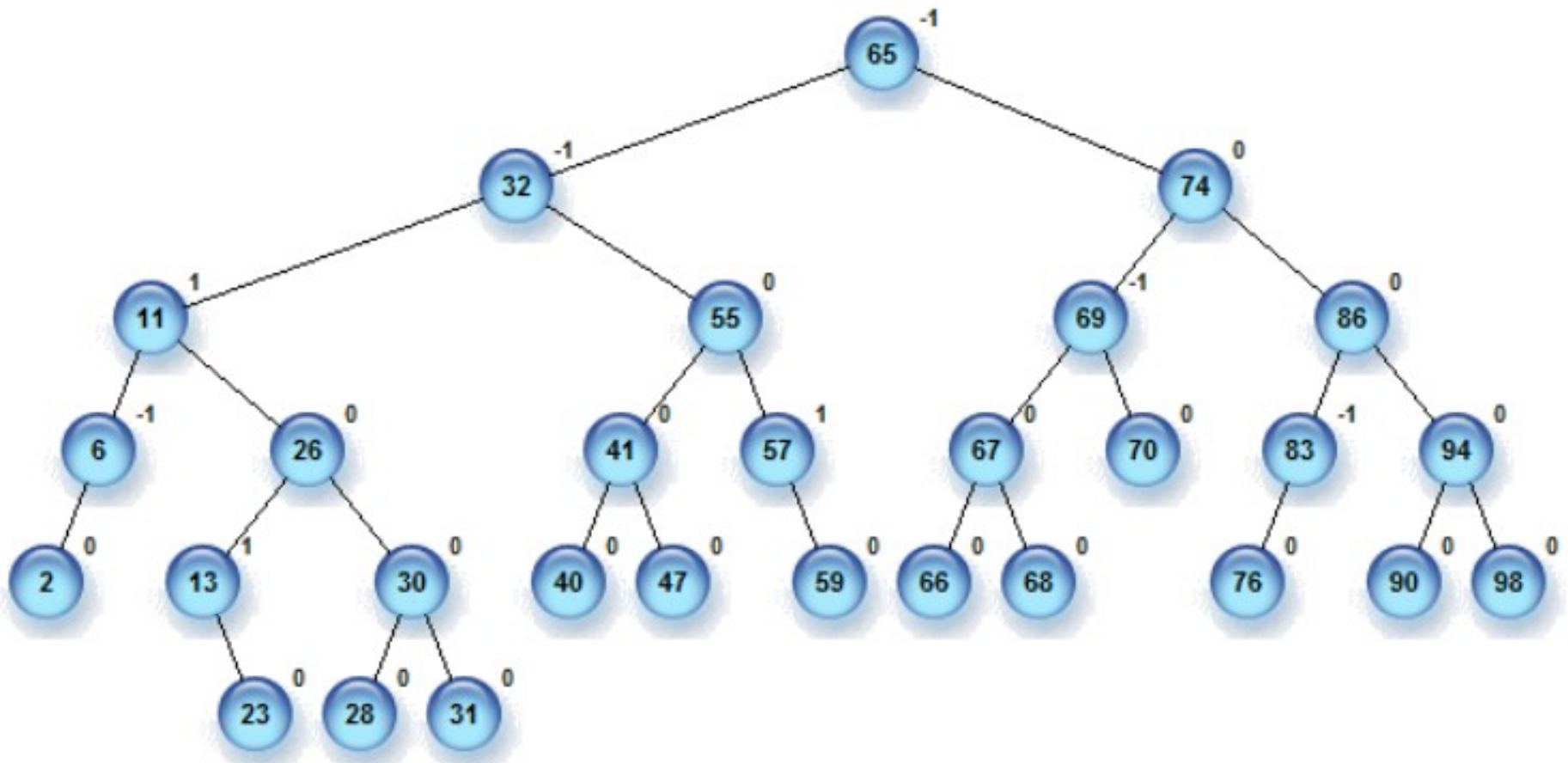
- Agregamos etapas a las operaciones de inserción y borrado
 - Árboles AVL
 - Árboles Rojo-Negro
 - Splay Trees

Árboles AVL

- Árboles binarios de búsqueda
- Propiedad de equilibrio AVL:
 - La altura del subárbol derecho y el izquierdo no difiere en mas de 1
- Factor de equilibrio:
 - $F_e(Nodo) = H(subarbol_{derecho}) - H(subarbol_{izquierdo})$
- Estructura ampliada con factor de equilibrio



Árboles AVL

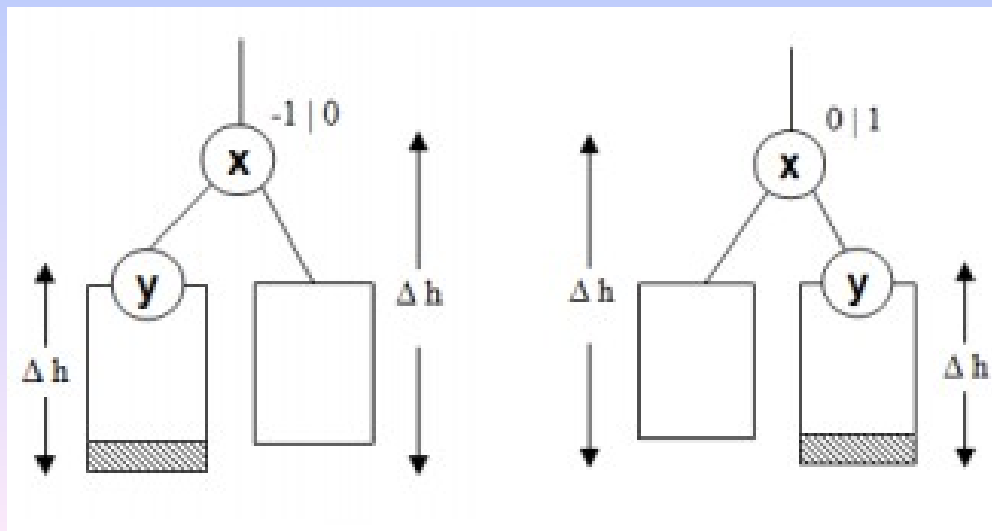


Árboles AVL

- Luego de insertar o borrar un nodo:
 - Solo los ascendientes pueden sufrir un cambio en su F_e (Solo en una unidad)
 - Se agrega una etapa de recorrido de los ascendientes
 - Recálculo de los F_e
 - Si alguno esta desequilibrado, se rota el árbol

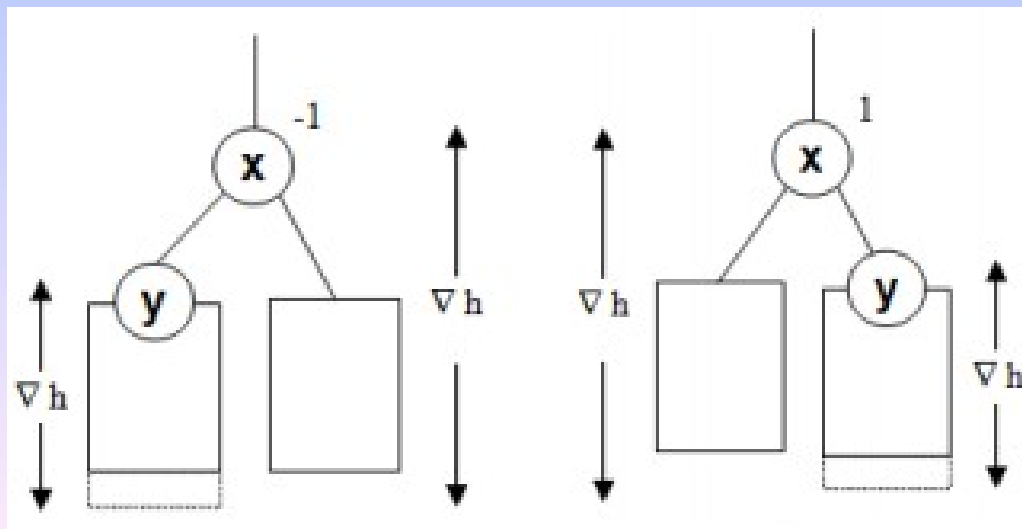
Árboles AVL

- Cambios en altura
 - Si un hijo incrementa su altura, el padre también la incrementa
 - Si su F_e era -1 o 0 (hijo izquierdo)
 - Si su F_e era 1 o 0 (hijo derecho)



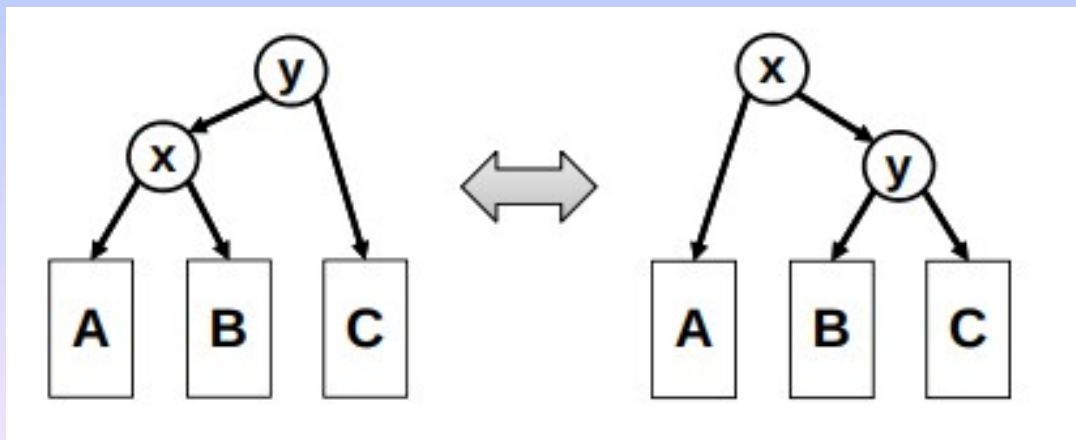
Árboles AVL

- Cambios en altura
 - Si un hijo decrementa su altura, el padre también la decrementa
 - Si su F_e era -1 (hijo izquierdo)
 - Si su F_e era 1 (hijo derecho)



Árboles AVL

- Rotación:
 - Reestructuración local de un subárbol para mantener propiedad de equilibrio



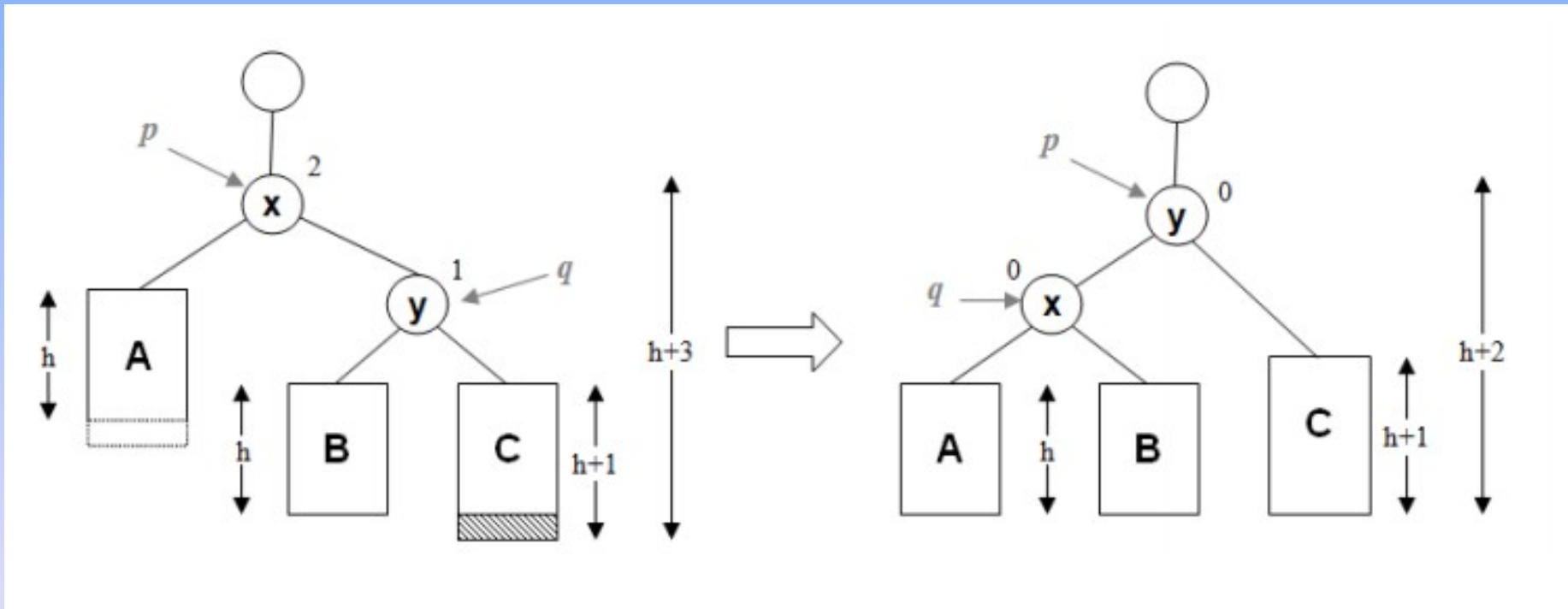
Árboles AVL

- Tipo de rotación:
 - Simples
 - Dobles
- Sentido de rotación
 - Izquierda
 - Derecha

Árboles AVL

- 6 casos posibles a considerar:
 - Rotación $2|1$ (simple derecha)
 - Rotación $2|0$ (simple derecha)
 - Rotación $2|-1$ (doble derecha)
 - Rotación $-2|-1$ (simple izquierda)
 - Rotación $-2|0$ (simple izquierda)
 - Rotación $-2|1$ (doble izquierda)

Árboles AVL



- Borrado en A que decrementa su altura (sin cambiar la de x)
- Inserción en C que incrementa su altura (cambia la de x e y)

Árboles binarios

- Implementemos los recorridos en el pizarrón