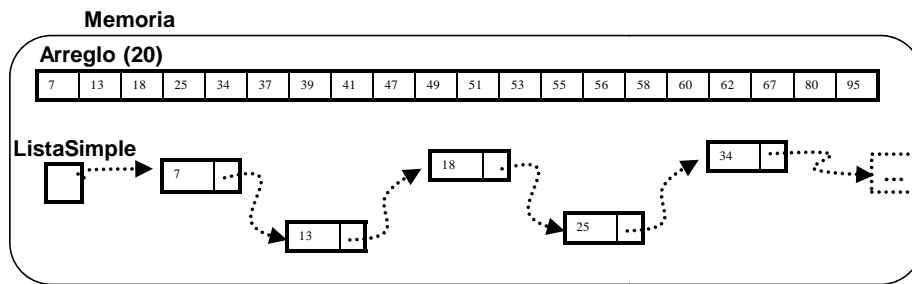


Arboles

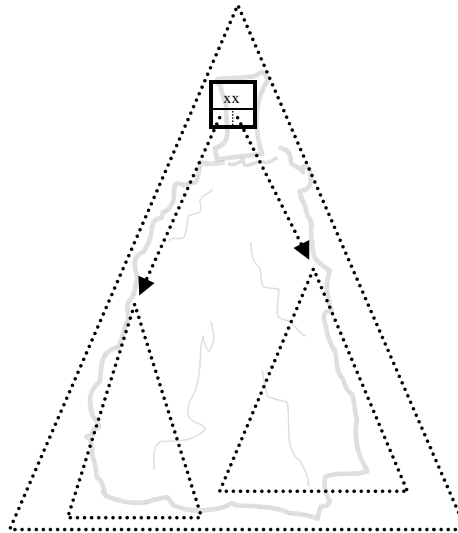
- Necesidad de una estructura con mayor velocidad de acceso.
- Estructura de árbol.
- Recorridos y accesos.
- Alta de un nodo en el árbol.
- Baja de un nodo en el árbol.
- Modificación de datos de un nodo del árbol.
- Árboles balanceados.

Costo de acceso a datos

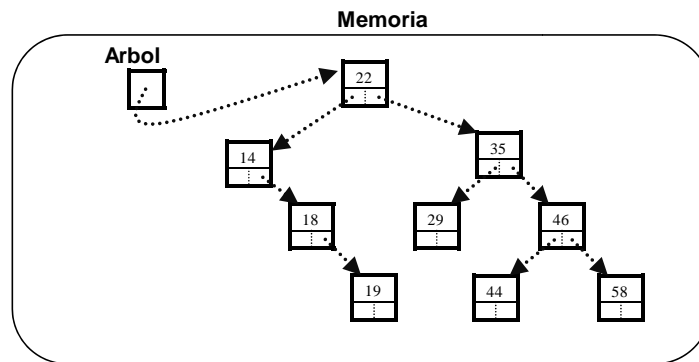


- ✓ En el arreglo, acceso por bisección.
(Costo máximo = $\log_2 N$)
(con 1.000 elementos = 10)
- ✓ En la lista, acceso secuencial.
(Costo promedio = $N/2$)
(con 1.000 elementos = 500)

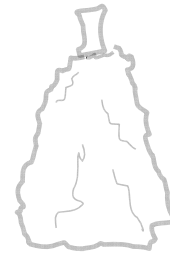
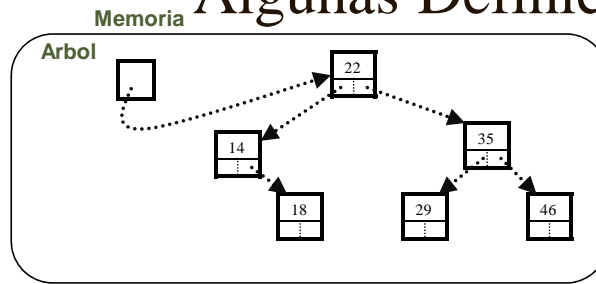
Árbol



Árbol binario

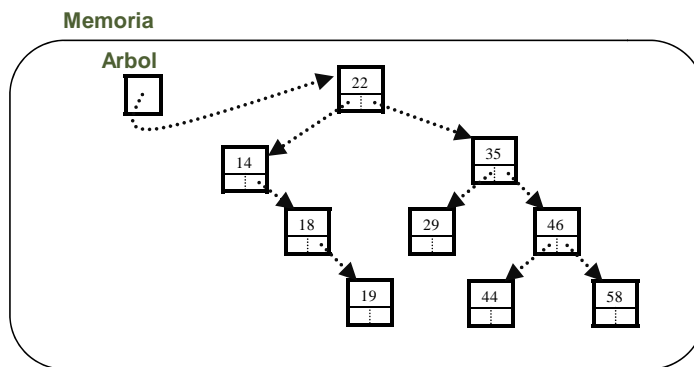


Algunas Definiciones



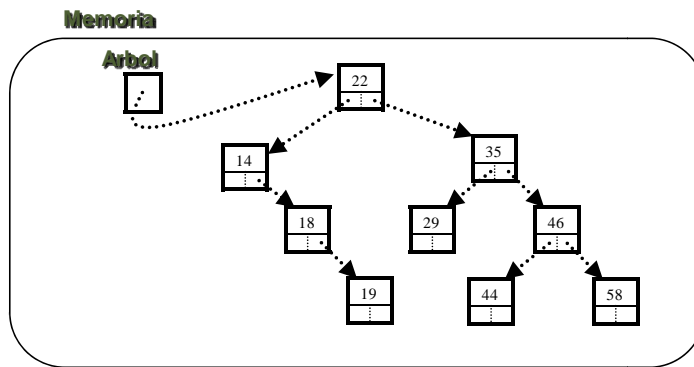
- ✓ **Grado** es el número máximo de hijos que tienen los nodos del árbol. Así, en el ejemplo anterior el árbol es de grado dos (binario). Por lo tanto una lista no es más que un árbol de grado uno.
- ✓ **Nodo Padre** de un nodo X es aquel que apunta al mismo. Cada nodo sólo puede tener un padre.
- ✓ **Nodo Hijo** de otro nodo Z es cualquier nodo apuntado por el nodo Z.
- ✓ **Nodo Raíz** es el único del árbol que no tiene padre.
- ✓ **Hojas** son todos los nodos que no tienen hijos.
- ✓ **Nodos Interiores** son los nodos que no son ni el nodo raíz, ni nodos hoja.
- ✓ **Camino** es una secuencia de nodos, en el que dos nodos consecutivos cualesquiera son padre e hijo.
- ✓ **Rama** es un camino desde el nodo raíz a una hoja.
- ✓ **Nivel** de un nodo, es el número de nodos del camino desde la raíz hasta dicho nodo. Nivel de raíz=1.
- ✓ **Altura** es el máximo número de nodos de las ramas del árbol.
- ✓ **Bosque** es un conjunto de árboles.

Definición de árbol



```
Type PuntArbol = ^TipoNodoArbol;  
TipoNodoArbol = record  
  Nro : Integer;  
  Menores, Mayores: PuntArbol  
End;  
Var ElArbol : PuntArbol;
```

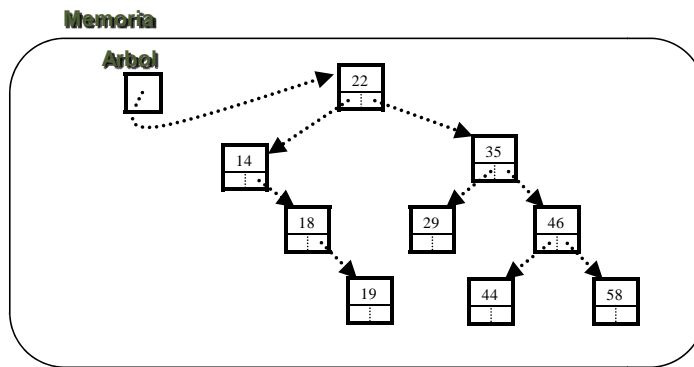
Recorrido en árbol



```
Procedure MostrarDatos(Pos: PuntArbol);
Begin
  if Pos <> nil then begin
    MostrarDatos(Pos ^.Menores);
    writeln(Pos ^.Nro);
    MostrarDatos(Pos ^.Mayores);
  end;
End;
```

**En orden
Ascendente**

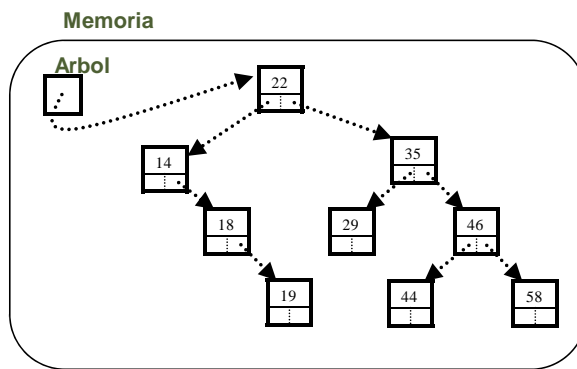
Recorrido en árbol



```
Procedure MostrarDatos(Pos: PuntArbol);  
Begin  
  if Pos <> nil then begin  
    MostrarDatos(Pos ^.Mayores);  
    writeln(Pos ^.Nro);  
    MostrarDatos(Pos ^.Menores);  
  end;  
End;
```

**En orden
Descendente**

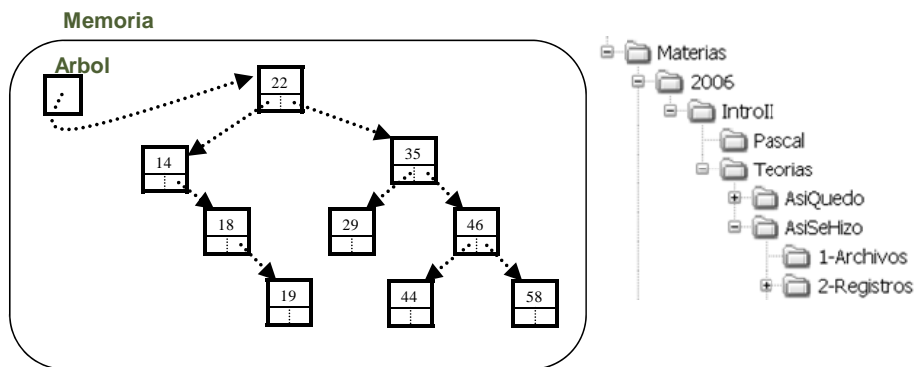
Modos de recorrido en árbol



```
Procedure MostrarDatos(Pos: PuntArbol);  
Begin  
  if Pos <> nil then begin  
    MostrarDatos(Pos ^.Menores);  
    writeln(Pos ^.Nro);  
    MostrarDatos(Pos ^.Mayores);  
  end;  
End;
```

In Order

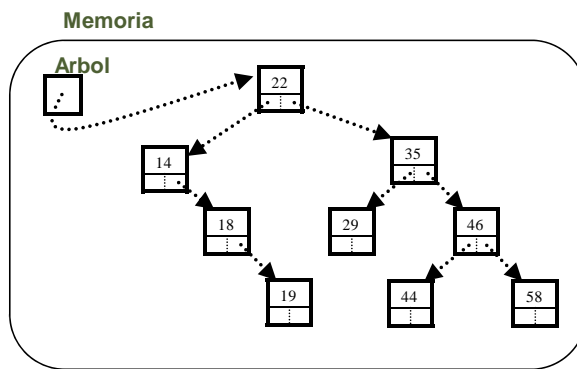
Modos de recorrido en árbol



```
Procedure MostrarDatos(Pos: PuntArbol);
Begin
  if Pos <> nil then begin
    writeln(Pos ^.Nro);
    MostrarDatos(Pos ^.Menores);
    MostrarDatos(Pos ^.Mayores);
  end;
End;
```

Pre Order

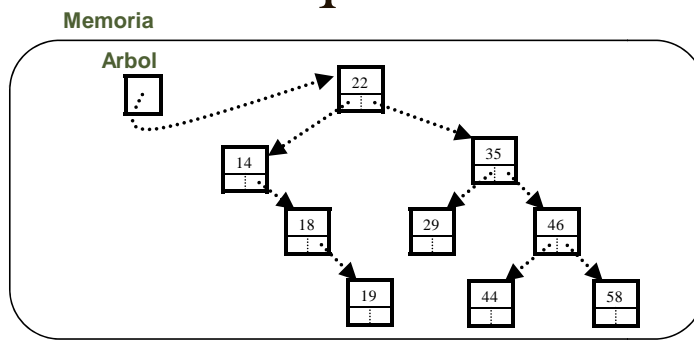
Modos de recorrido en árbol



```
Procedure MostrarDatos(Pos: PuntArbol);
Begin
  if Pos <> nil then begin
    MostrarDatos(Pos ^.Menores);
    MostrarDatos(Pos ^.Mayores);
    writeln(Pos ^.Nro);
  end;
End;
```

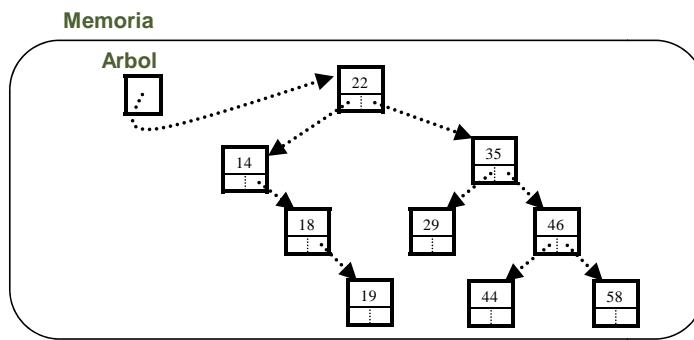
Post Order

Búsqueda en árbol



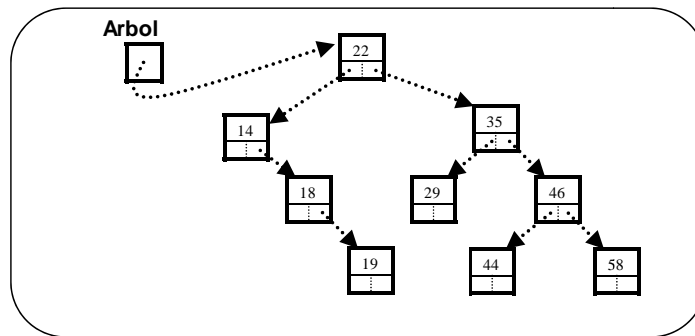
```
Function Puntero(Pos: PuntArbol, Valor: Integer):PuntArbol;
Begin
  if Pos = nil then
    Puntero := nil
  else if Pos^.Nro = Valor then
    Puntero := Pos
  else if Pos^.Nro < Valor then
    Puntero := Puntero (Pos^.Mayores, Valor);
  else
    Puntero := Puntero (Pos^.Menores, Valor);
End;
```

Alta en árbol



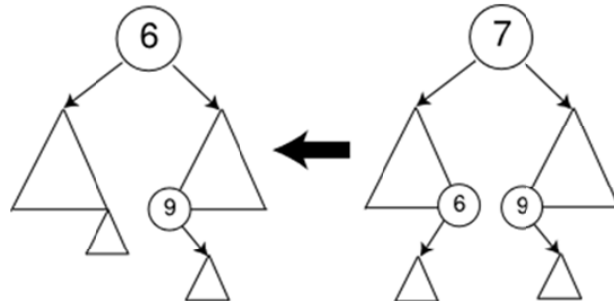
```
Procedure Alta(var Pos: PuntArbol, Valor: Integer);
Begin
  if Pos = nil then begin
    new(Pos);
    Pos^.Nro := Valor;
    Pos^.Menores := nil; Pos^.Mayores := nil;
  end else if Pos^.Nro < Valor then
    Alta (Pos^.Mayores, Valor);
  else
    Alta (Pos^.Menores, Valor);
End;
```

Baja en árbol

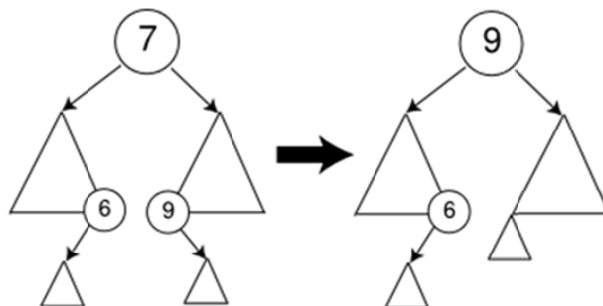


- ✓ **Borrar un nodo sin hijos ó nodo hoja:** simplemente se borra y se establece a nulo el apuntador de su padre.
- ✓ **Borrar un nodo con un subárbol hijo:** se borra el nodo y se asigna su subárbol hijo como subárbol de su padre.
- ✓ **Borrar un nodo con dos subárboles hijo:** es el caso complejo que veremos a continuación.

Baja en árbol



Baja en árbol



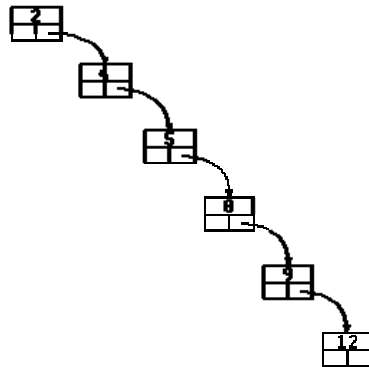
Modificación de datos de árbol

✓ ***Dato no Clave:*** No hay nada a modificar.

✓**Dato Clave:** Si modifica su posición se puede dar de baja y alta en la nueva posición.

Idea sobre balanceo de árbol

Peor situación:



Mejor situación:

