Árboles:

Búsqueda acotada
Borrado

Recorrido acotado en un ABB

Hemos analizado algunas situaciones que obligan a recorrer todos los nodos del árbol, por ejemplo imprimir los nodos del árbol

Cuando necesitamos mostrar los datos que están comprendidos entre **dos valores determinados dentro del orden del árbol** ¿cómo lo resolvemos?

Retomando…

```
Procedure preOrden( a: arbol );
begin
  if ( a <> nil ) then begin
    writeln (a^.dato);
    preOrden (a^.HI);
    preOrden (a^.HD)
  end;
end;
```

```
ImprimirAcotado(arbol, inf, sup);

  si arbol no está vacío
     si el valor en arbol es > = inf and
          el valor en arbol es < = sup
               mostrar valor
     ImprimirAcotado (hijo_izq_arbol, inf,
);
     ImprimirAcotado (hijo_der_arbol, inf,
);
```

Recorrido acotado en un ABB

```
recorridoAcotado(arbol, inf, sup);

si arbol no está vacío
  si el valor en arbol es > = inf and el valor en arbol es < =
 sup
     mostrar valor
     recorridoAcotado (hijo_izq_arbol, inf, sup);
     recorridoAcotado (hijo_der_arbol, inf, sup);
  si el valor en arbol es > sup
     recorridoAcotado (a^.izq, inf, sup);
  si el valor en arbol es < inf
     recorridoAcotado (a^.der, inf, sup);
```

¿se puede mejorar esta solución para evitar comparaciones ?

Recorrido acotado en un ABB

```
Procedure busquedaAcotada(a: Arbol_Usuarios; inf:integer;
sup:integer);
begin
  if (a <> nil) then
    if (a^.dato.id >= inf) then
      if (a^.dato.id <= sup) then begin
        write(a^.dato.nombre);
        busquedaAcotada(a^.hi, inf, sup);
        busquedaAcotada (a^.hd, inf, sup);
      end
      else
        busquedaAcotada(a^.hi, inf, sup)
    else
      busquedaAcotada(a^.hd, inf, sup);
end;
```

Actividades en Máquina

•Utilizando el **ProgramaGenerarArbol (de la clase anterior)** realice las siguientes actividades:

•**ACTIVIDAD 7**

a) Generar un árbol de números enteros utilizando la lista ya creada.

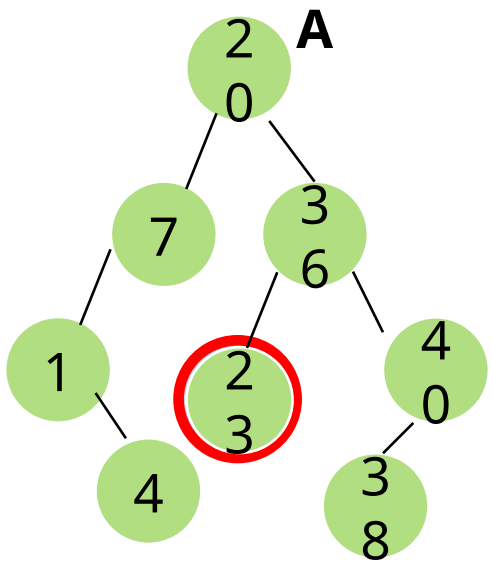
b) Mostrar el contenido del árbol en forma creciente.

c)Implementar el módulo verValoresEnRango que reciba un árbol y dos valores, que indiquen un rango, e informe los valores del árbol que se encuentren en dicho **rango**.

d) Invocar al módulo verValoresEnRango con dos valores leídos de teclados.

Borrar un elemento en un ABB

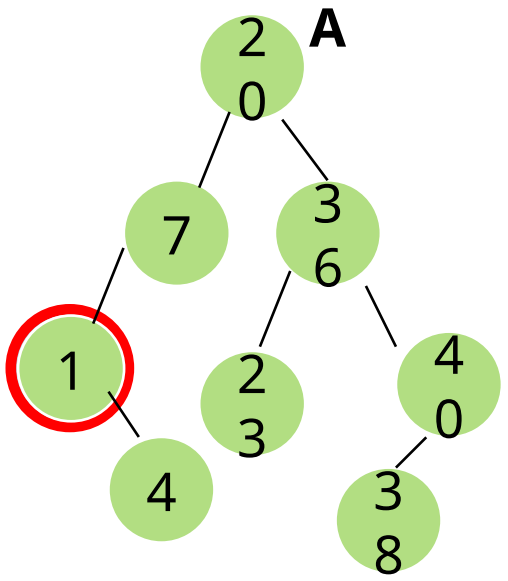
Se deben considerar diferentes situaciones:



1. Si el nodo **es una hoja**

✔️Se puede borrar inmediatamente (actualizando direcciones)

Borrar un elemento en un ABB

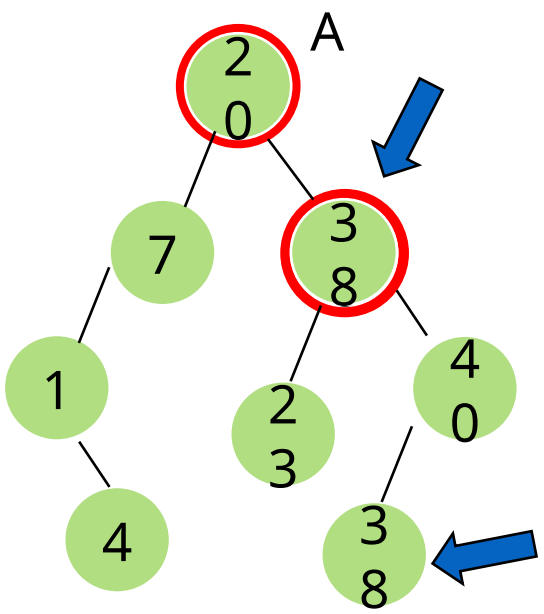


2. Si el nodo tiene un hijo

✔Si **el nodo tiene un hijo**, el nodo puede ser borrado después que su padre actualice el puntero al hijo del nodo que se quiere borrar.

Borrar un elemento en un ABB

3. Si el nodo tiene dos hijos



Se asume una estrategia…

1. Se busca el valor a borrar (ej 36).

2. Se busca y selecciona el hijo mas a la izquierda del subárbol derecho del nodo a borrar (o el hijo mas a la derecha del subárbol izquierdo). ¿Por qué?

3. Se intercambia el valor del nodo encontrado por el que se quiere borrar

4. Se llama al borrar a partir del hijo derecho con el valor del nodo encontrado. ¿Qué característica tiene ese nodo encontrado?

```
borrarElemento (árbol, dato, resultado)
  si árbol es vacío no se encontró el dato a borrar
  sino
    si el dato en árbol es > dato
      borrarElemento (hijo_izquierdo_del_árbol, dato, resultado);
    sino
      si el dato en árbol es < dato
        borrarElemento (hijo_derecho_del_árbol, dato, resultado);
      sino
        se encontró el dato a borrar
              si no tiene ningún hijo
              sino
          si tiene sólo hijo derecho …
          sino {tiene los 2 hijos}
            si tiene sólo hijo izquierdo …
             sino
               buscar el mínimo del hijo derecho
               reemplazar el valor del árbol por el mínimo
               borrarElemento
(hijo_derecho_del_árbol,mínimo,resultado);
```

Actividades en Máquina

•Renombrar **ProgramaGenerarArbol** como **programaBorrarEnArbol** y realice las siguientes actividades:

•**ACTIVIDAD 8**

a) Implementar el módulo borrarElemento que reciba un árbol y un valor a eliminar.

b) Invocar al módulo borrarElemento con un valor que se ingresa de teclado.

c) Invocar al módulo imprimirpornivel con el árbol resultante en el punto b).