



# Taller de Programación



# AGENDA



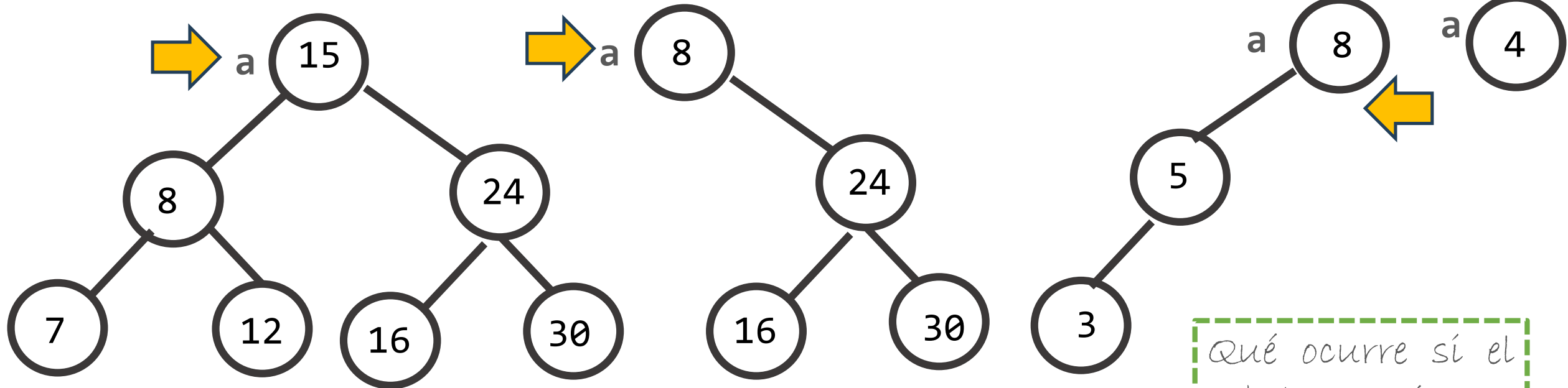
Esturctura de datos ABB – MAYOR ELEMENTO

Esturctura de datos ABB – MENOR ELEMENTO



# ARBOLES BINARIOS DE BUSQUEDA - MENOR

Supongamos que disponemos de un árbol binario de búsqueda, y queremos implementar un módulo que retorne el valor más pequeño que está cargado en el árbol.



El valor más pequeño de cualquier ABB es el valor ubicado "mas" a la izquierda

Qué ocurre si el árbol es vacío?

Cómo lo implemento?



# ARBOLES BINARIOS DE BUSQUEDA - MENOR

DEVUELVE EL VALOR MINIMO

Programa arbolesEnteros;

Type

arbol = ^nodo;

nodo = record

dato: integer;

HI: arbol;

HD: arbol;

end;

Var

**a:arbol; min:integer;**

Begin

cargarArbol(a);

if (a <> nil) then

min:= minimo(a);

write (min);

End.

DEVUELVE EL NODO QUE CONTIENE EL MINIMO

Programa arbolesEnteros;

Type

arbol = ^nodo;

nodo = record

dato: integer;

HI: arbol;

HD: arbol;

end;

Var

**a:arbol; min:arbol;**

Begin

cargarArbol(a);

min:= minimoNodo(a);

if (min <> nil) then write (min^.dato);

End.



# ARBOLES BINARIOS DE BUSQUEDA - MENOR

DEVUELVE EL VALOR MINIMO

Programa arbolesEnteros;

Type

arbol = ^nodo;

nodo = record

dato: integer;

HI: arbol;

HD: arbol;

end;

Var

a:arbol; min:integer;

Begin

cargarArbol(a);

if (a <> nil) then

min:= minimo(a);

write (min);

End.

```
function minimo (a:arbol): integer;
```

```
begin
```

```
  if (a^.HI = nil) then
```

```
    minimo:= a^.dato
```

```
  else mínimo:= mínimo (a^.HI);
```

```
end;
```

Puede implementarse  
de manera iterativa?

```
function minimo (a:arbol): integer;
```

```
begin
```

```
  while (a^.HI <> nil) do
```

```
    a:= a^.HI;
```

```
    minimo:= a^.dato
```

```
end;
```



# ARBOLES BINARIOS DE BUSQUEDA - MENOR

DEVUELVE EL NODO QUE CONTIENE EL MINIMO

Programa arbolesEnteros;

Type

arbol = ^nodo;

nodo = record

dato: integer;

HI: arbol;

HD: arbol;

end;

Var

a:arbol; min:arbol;

Begin

cargarArbol(a);

min:= minimoNodo(a);

if (min <> nil) then write (min^.dato);

End.

function minimoNodo (a:arbol): arbol;

Begin

if (a = nil) then minimoNodo:= nil

else if (a^.HI = nil) then

minimoNodo:= a;

else minimoNodo:= minimoNodo (a^.HI);

end;

Puede implementarse  
de manera iterativa?

function minimoNodo (a:arbol): arbol;

Begin

if (a = nil) then minimoNodo:= nil

else

while (a^.HI <> nil) do

a:= a^.HI;

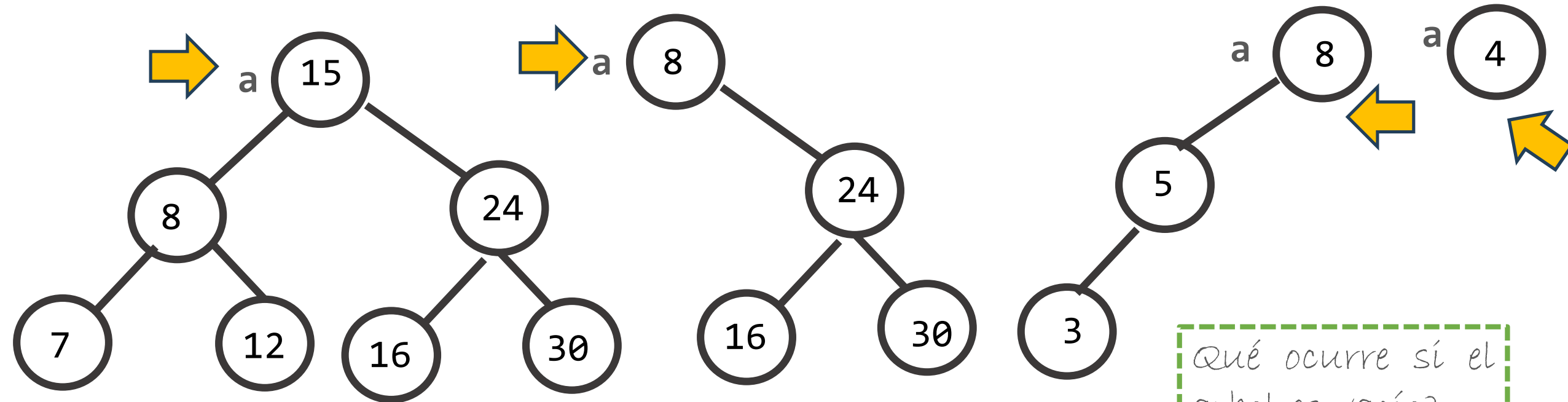
minimoNodo:= a

end;



# ARBOLES BINARIOS DE BUSQUEDA - MAXIMO

Supongamos que disponemos de un árbol binario de búsqueda, y queremos implementar un módulo que retorne el valor más grande que está cargado en el árbol.



El valor más grande de cualquier ABB es el valor ubicado "mas " a la derecha

Qué ocurre si el árbol es vacío?

Cómo lo implemento?



# ARBOLES BINARIOS DE BUSQUEDA - MAXIMO

DEVUELVE EL VALOR MAXIMO

```
Programa arbolesEnteros;
```

```
Type
```

```
    arbol = ^nodo;
```

```
    nodo = record
```

```
        dato: integer;
```

```
        HI: arbol;
```

```
        HD: arbol;
```

```
    end;
```

```
Var
```

```
    a:arbol; max:integer;
```

```
Begin
```

```
    cargarArbol(a);
```

```
    if (a <> nil) then
```

```
        max:= maximo(a);
```

```
        write (max);
```

```
End.
```

DEVUELVE EL NODO QUE CONTIENE EL MAXIMO

```
Programa arbolesEnteros;
```

```
Type
```

```
    arbol = ^nodo;
```

```
    nodo = record
```

```
        dato: integer;
```

```
        HI: arbol;
```

```
        HD: arbol;
```

```
    end;
```

```
Var
```

```
    a:arbol; max:arbol;
```

```
Begin
```

```
    cargarArbol(a);
```

```
    max:= maximoNodo(a);
```

```
    if (max <> nil) then    write (max^.dato);
```

```
End.
```





# ARBOLES BINARIOS DE BUSQUEDA - MAXIMO

DEVUELVE EL VALOR MAXIMO

```
Programa arbolesEnteros;
Type
  arbol = ^nodo;
  nodo = record
    dato: integer;
    HI: arbol;
    HD: arbol;
  end;
Var
  a:arbol; max:integer;

Begin
  cargarArbol(a);
  if (a <> nil) then
    max:= maximo(a);
    write (max);
  End.
```

```
function maximo (a:arbol): integer;
```

```
begin
  if (a^.HD = nil) then
    maximo:= a^.dato

    else maximo:= maximo (a^.HD);
end;
```

Puede implementarse  
de manera iterativa?

```
function maximo (a:arbol): integer;
```

```
begin
  while (a^.HD <> nil) do
    a:= a^.HD;
    maximo:= a^.dato;
end;
```



# ARBOLES BINARIOS DE BUSQUEDA - MAXIMO

DEVUELVE EL NODO QUE CONTIENE EL MAXIMO

Programa arbolesEnteros;

Type

arbol = ^nodo;

nodo = record

dato: integer;

HI: arbol;

HD: arbol;

end;

Var

a:arbol; max:arbol;

Begin

cargarArbol(a);

max:= maximoNodo(a);

if (max <> nil) then write (max^.dato);

End.

function maximoNodo (a:arbol): arbol;

Begin

if (a = nil) then maximoNodo:= nil

else if (a^.HD = nil) then

maximoNodo:= a

else maximoNodo:= maximoNodo (a^.HD);

end;

Puede implementarse  
de manera iterativa?

function maximoNodo (a:arbol): arbol;

Begin

if (a = nil) then maximoNodo:= nil

else

while (a^.HD <> nil) do

a:= a^.HD;

maximoNodo:= a;

end;