



Taller de Programación



AGENDA

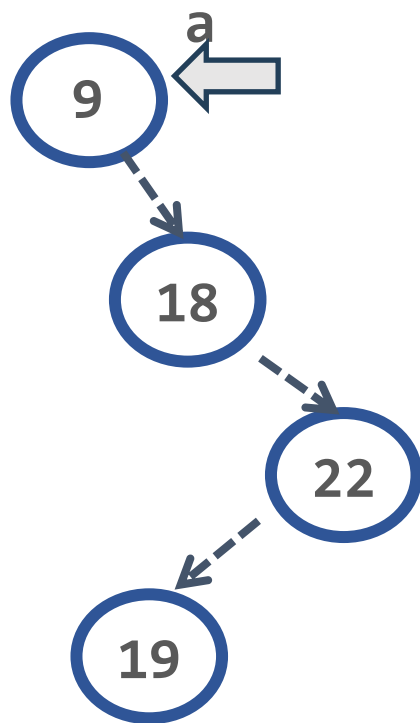


Operaciones ABB- RECORRIDOS

BUSQUEDA



ESTRUCTURA DE DATOS ARBOL - RECORRIDO



Para poder recorrer un ABB siempre debe comenzarse el recorrido desde la raíz.

Una vez que se esta parado en la raíz debe hacerse la acción que se quiera con el valor (imprimir, agregar en otro arbol, agregarlo en una lista, modificarlo, etc).

Luego debe tomarse uno de sus hijos y realizar la misma acción que para el nodo padre y luego el otro de sus hijos.

Cuál es el caso base?

Cuántos llamados recursivos se hacen en cada nodo?

Cómo se implementa?



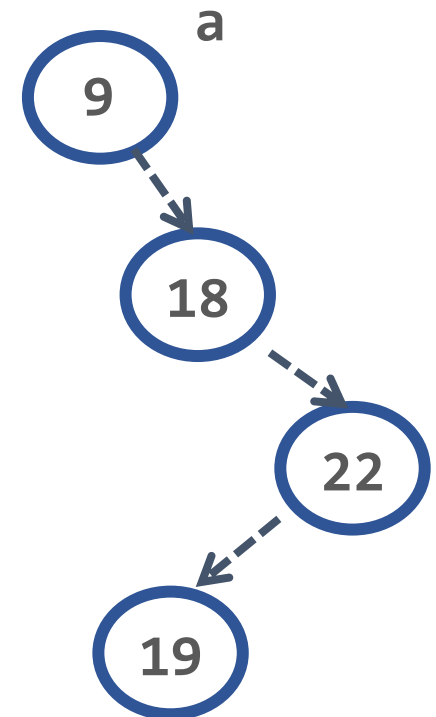
ESTRUCTURA DE DATOS ARBOL - RECORRIDO

```
Programa arboles;  
Type  
  arbol = ^nodo;  
  nodo = record  
    dato: integer;  
    HI: arbol;  
    HD: arbol;  
  end;  
  
Var  
  a:arbol;  num:integer;  
  
Begin  
  a:= nil;  
  read (num);  
  while (num <> 50) do  
    begin  
      agregar (a,num);  
      read (num);  
    end;  
    recorrido_enOrden(a);  
End.
```

```
Procedure enOrden ( a : arbol );  
begin  
  if ( a <> nil ) then begin  
    enOrden (a^.HI);  
    write (a^.dato); //o cualquier otra acción  
    enOrden (a^.HD);  
  end;  
end;
```

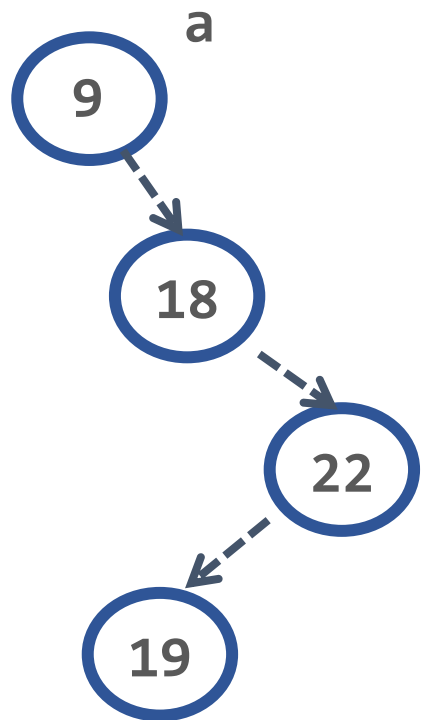
Es lo mismo pasar
a por referencia?

Cómo funciona?



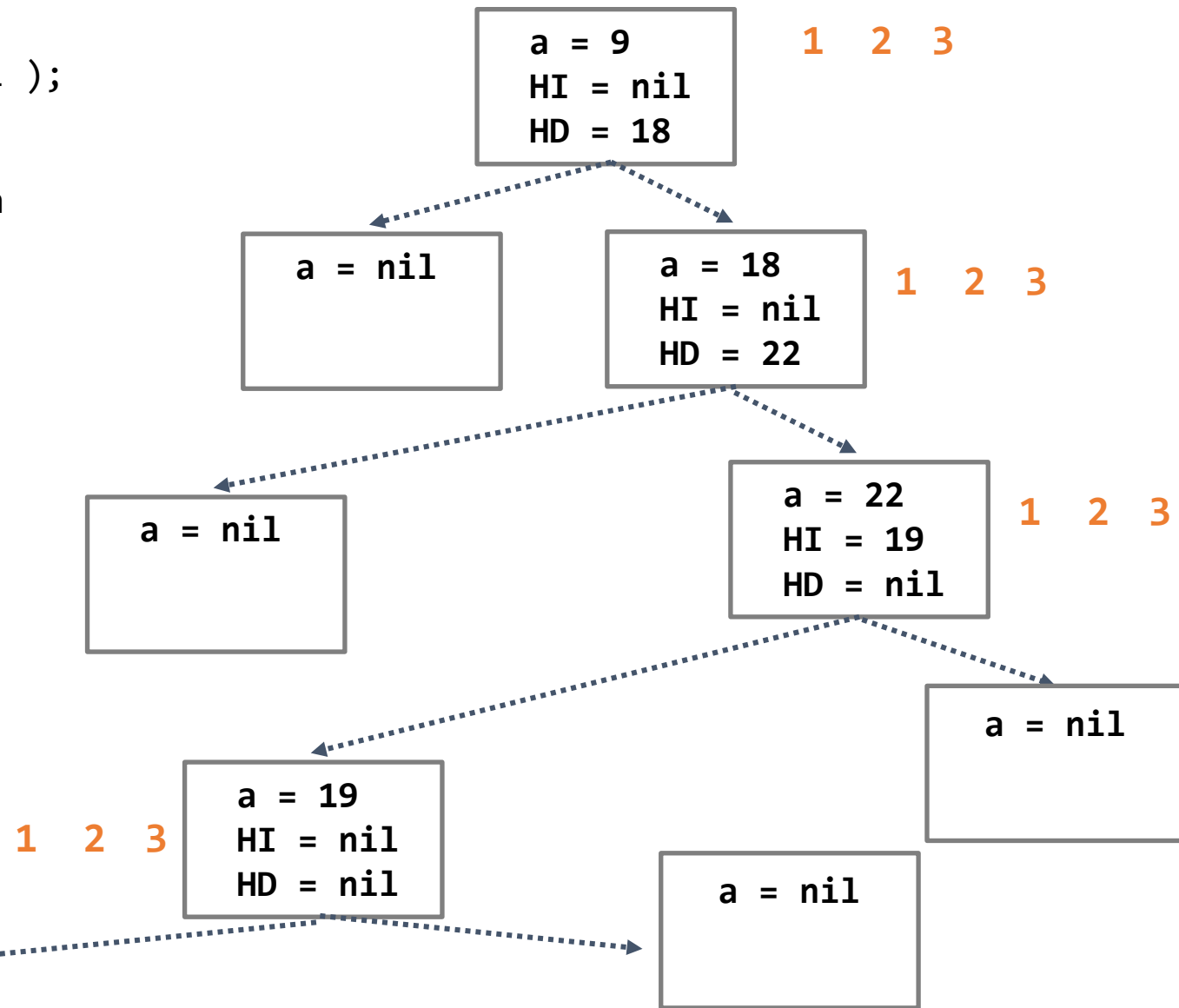


ESTRUCTURA DE DATOS ARBOL - RECORRIDO



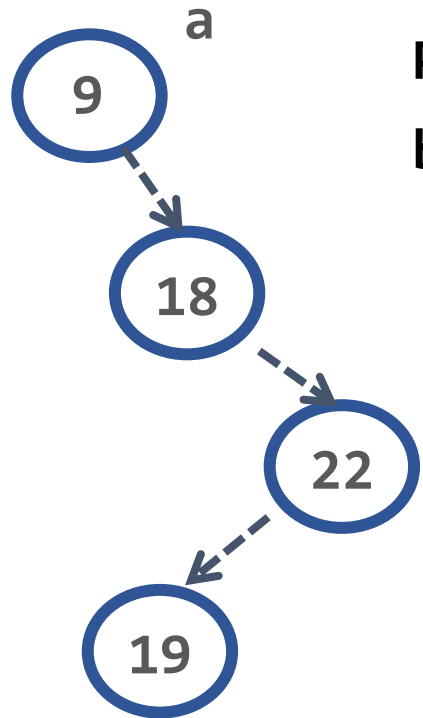
```
Procedure enOrden ( a : arbol );  
begin  
  if ( a <> nil ) then begin  
    1 enOrden (a^.HI);  
    2 write (a^.dato);  
    3 enOrden (a^.HD);  
  end;  
end;
```

Pantalla 9
Pantalla 18
Pantalla 19
Pantalla 22





ESTRUCTURA DE DATOS ARBOL - RECORRIDO



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

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

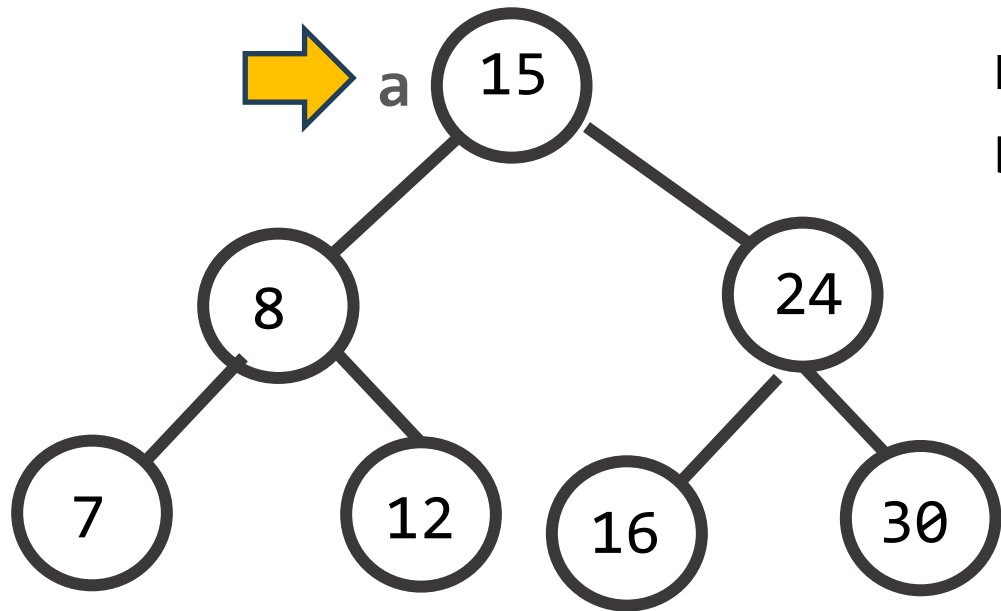
Qué imprímen?

Si *a* se pasa por
referencia que
imprime cada uno?



ARBOLES BINARIOS DE BUSQUEDA - BUSCAR

Supongamos que disponemos de un árbol binario de búsqueda, y queremos implementar un módulo que retorne un booleano si se encuentra un valor buscado que se recibe como parámetro.



```
Procedure Buscar (a:arbol;x:integer;var ok:boolean);  
begin  
  if ( a <> nil ) then begin  
    buscar (a^.HI,x,ok);  
    if (a^.dato = x) then ok:= true;  
    buscar (a^.HD,x,ok);  
  end;  
end;
```

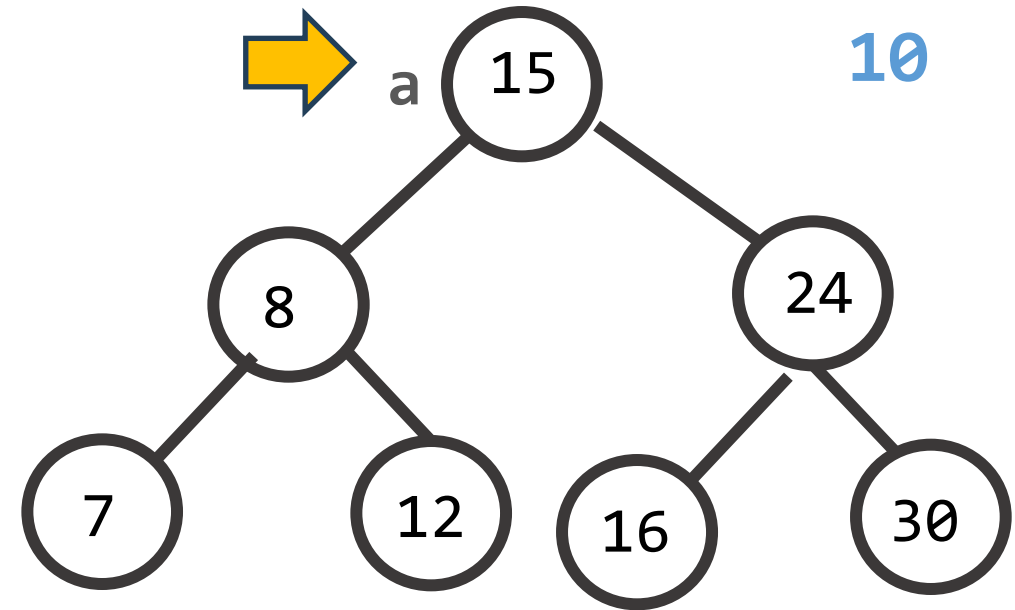
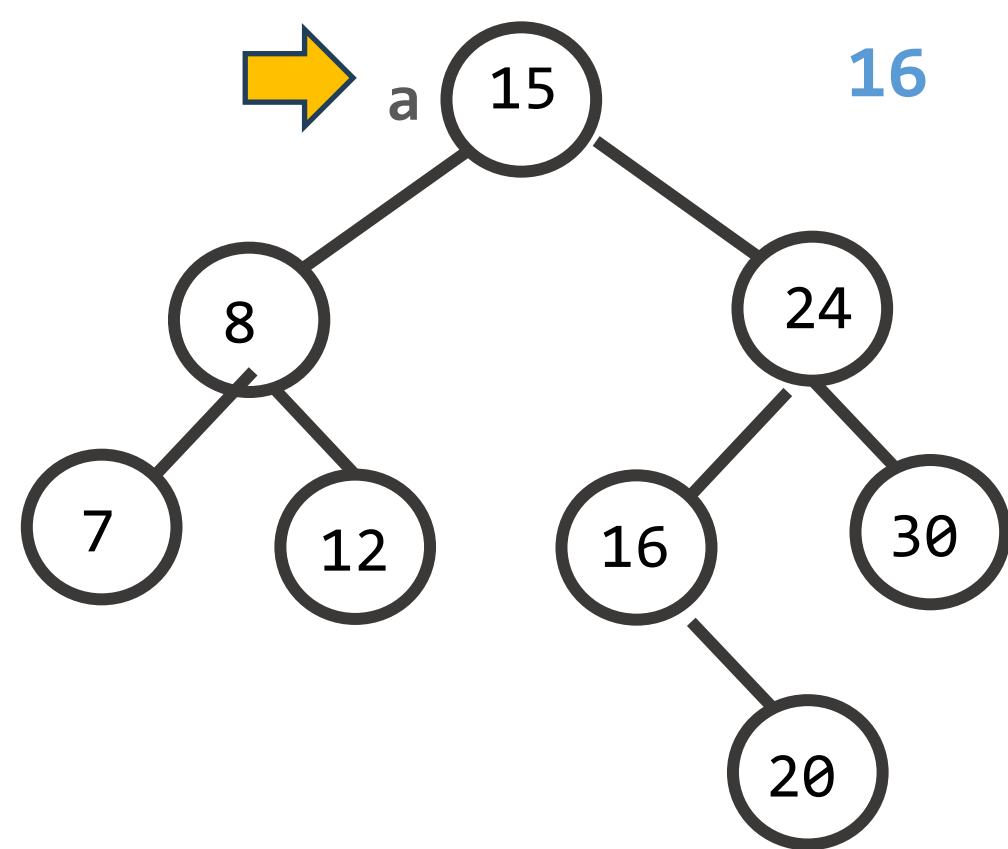


Para buscar un valor siempre se debe "aprovechar" el orden del ABB.



ARBOLES BINARIOS DE BUSQUEDA - BUSCAR

Supongamos que disponemos de un árbol binario de búsqueda, y queremos implementar un módulo que retorne el nodo donde se encuentra un valor buscado que se recibe como parámetro.



Cómo lo implemento?



ARBOLES BINARIOS DE BUSQUEDA - BUSCAR

DEVUELVE EL VALOR BOOLEAN

Programa arbolesEnteros;

Type

arbol = ^nodo;

nodo = record

dato: integer;

HI: arbol;

HD: arbol;

end;

Var

a:arbol;min:integer;ok:booleanx:integer

Begin

cargarArbol(a); read(x);

if (a <> nil) then

ok:= buscar(a,x);

write (ok);

End.

DEVUELVE EL NODO QUE CONTIENE EL BUSCADO

Programa arbolesEnteros;

Type

arbol = ^nodo;

nodo = record

dato: integer;

HI: arbol;

HD: arbol;

end;

Var

a:arbol; bus:arbol; x:integer;

Begin

cargarArbol(a); read(x);

bus:= buscarNodo(a,x);

if (bus <> nil) then write ("encontro");

End.



ARBOLES BINARIOS DE BUSQUEDA - BUSCAR

DEVUELVE EL BOOLEAN

```
Programa arbolesEnteros;
```

```
Type
```

```
    arbol = ^nodo;
```

```
    nodo = record
```

```
        dato: integer;
```

```
        HI: arbol;
```

```
        HD: arbol;
```

```
    end;
```

```
Var
```

```
    a:arbol; min:integer; x:integer;
```

```
Begin
```

```
    cargarArbol(a); read(x);
```

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

```
        ok:= buscar(a,x);
```

```
        write (ok);
```

```
End.
```

```
Function buscar (a:arbol; x:integer): boolean;
```

```
begin
```

```
    if (a = nil) then buscar:= false
```

```
    else (a^.dato = x) then buscar:= true
```

```
    else if (x > a^.dato) then buscar:= buscar(a^.HD, x)
```

```
    else buscar:= buscar(a^.HI, x)
```

```
end;
```

Puede implementarse
de manera iterativa?



ARBOLES BINARIOS DE BUSQUEDA - BUSCAR

DEVUELVE EL NODO BUSCADO

Programa arbolesEnteros;

Type

arbol = ^nodo;

nodo = record

dato: integer;

HI: arbol;

HD: arbol;

end;

Var

a:arbol; bus:arbol; x:integer;

Begin

cargarArbol(a); read(x);

bus:= buscarNodo(a,x);

if (bus <> nil) then write ("encontro");

End.

function buscarNodo (a:arbol; x:integer): arbol;

Begin

if (a = nil) then buscarNodo:= nil

else (a^.dato = x) then buscarNodo:= a

else if (x > a^.dato) then

buscarNodo:= buscarNodo(a^.HD, x)

else buscarNodo:= buscarNodo(a^.HI, x);

End;

Puede implementarse
de manera iterativa?



ARBOLES BINARIOS DE BUSQUEDA -



Operaciones:

- Buscar valores dentro de un rango
- Obtener el valor máximo de un ABB
- Obtener el valor mínimo de un ABB
- Calcular la cantidad de nodos que tiene un ABB
- Calcular el nivel de un ABB
- Calcular la altura de un ABB