



Taller de Programación



AGENDA

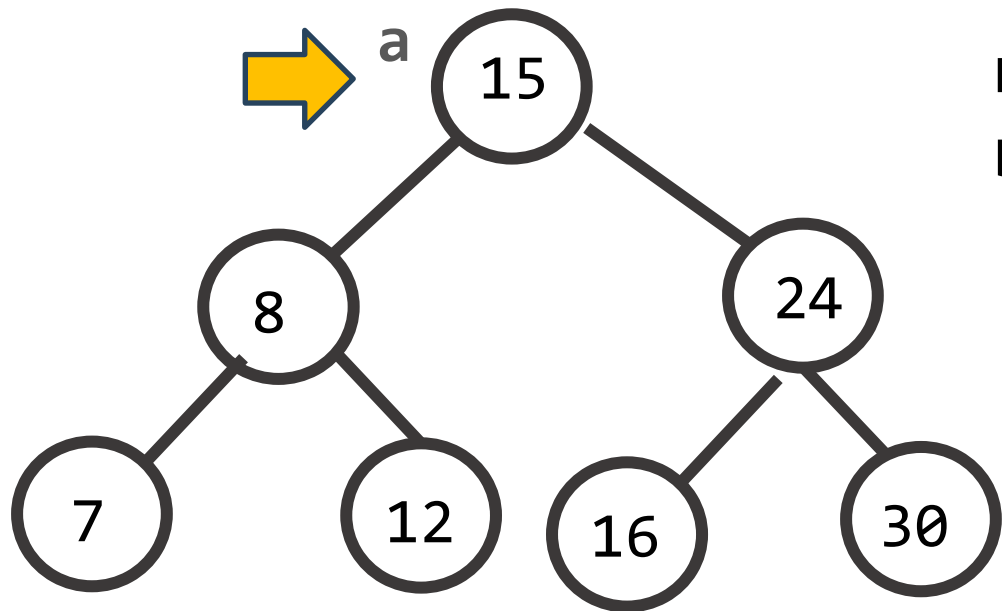


Esturctura de datos ABB – BUSCAR UN VALOR



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;
```

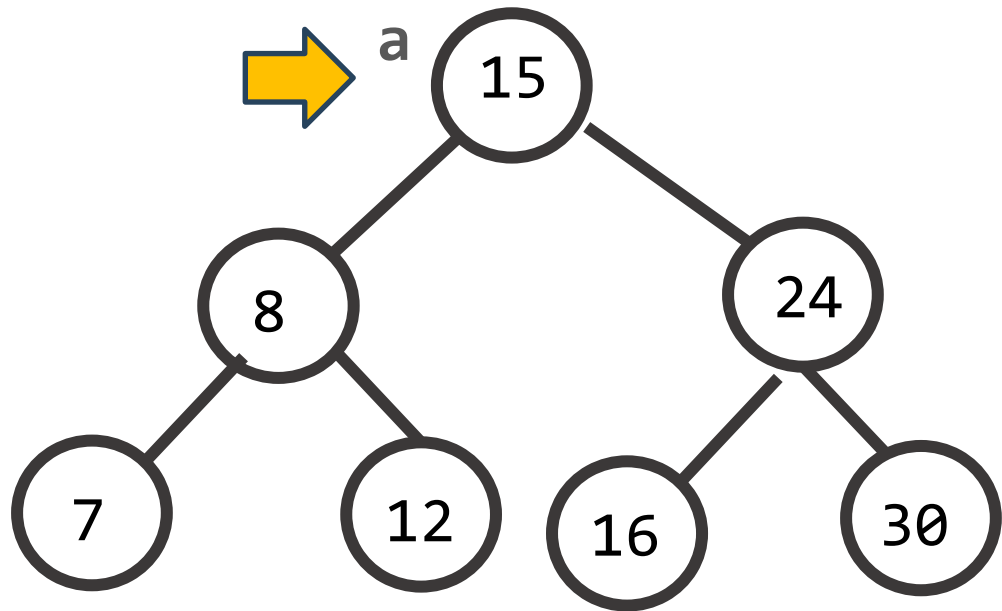


Para buscar un valor siempre se debe "aprovechar" el orden del ABB.



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.



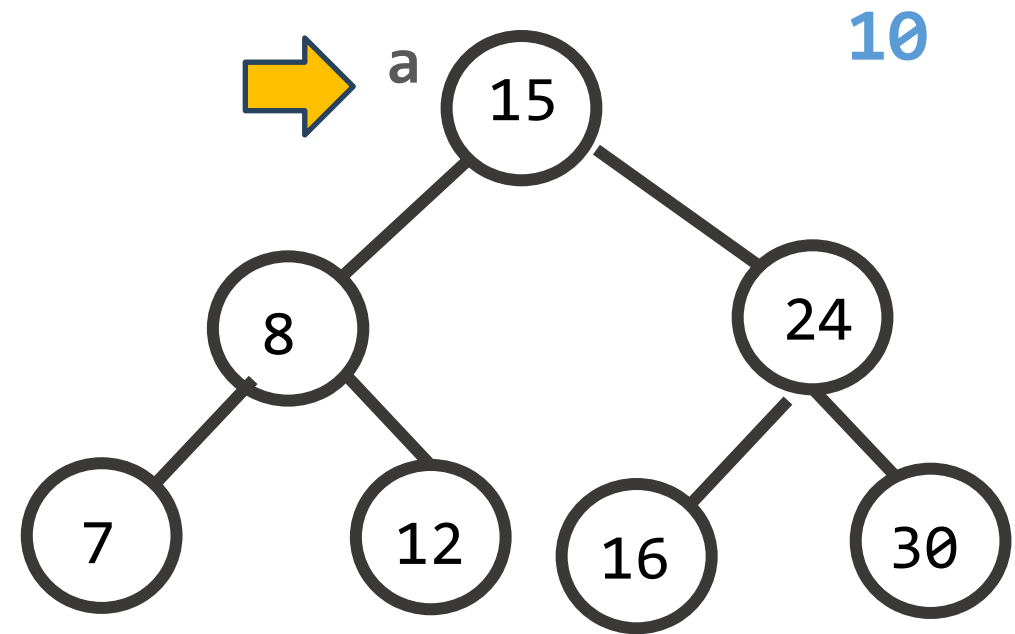
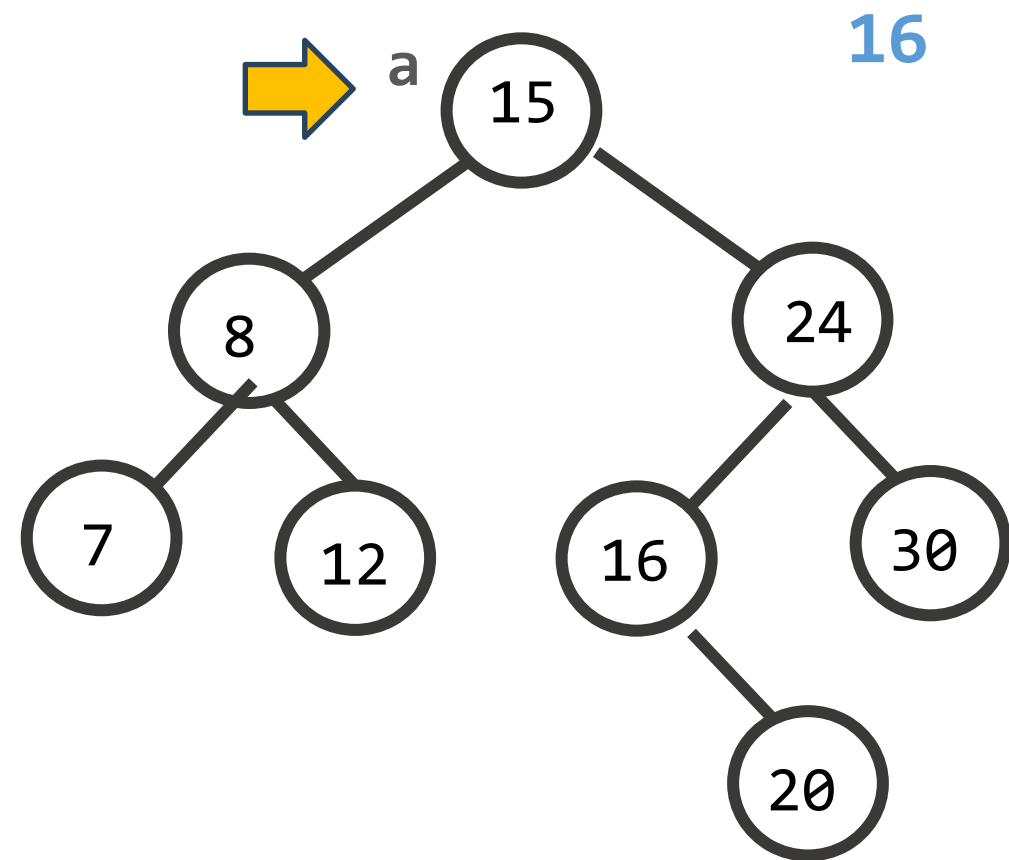
```
Procedure Buscar (a:arbol;x:integer;var ab:arbol);  
begin  
  if ( a <> nil ) then begin  
    buscar (a^.HI,x,ab);  
    if (a^.dato = x) then ab:= a;  
    buscar (a^.HD,x,ab);  
  end;  
end;
```





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?



ÁRBOLES – OTRAS OPERACIONES

Imprimir los valores entre rangos

Calcular el nivel de un arbol

Calcular la altura de un arbol

Obtener un árbol espejo