



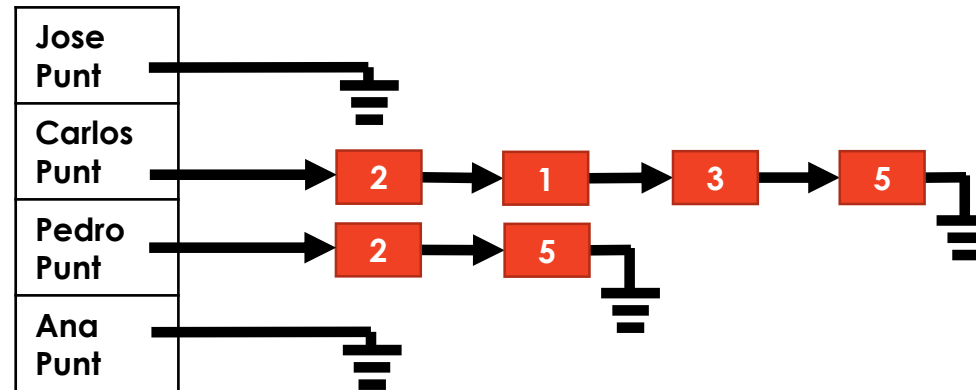
Estructuras combinadas

Introducción a la programación II

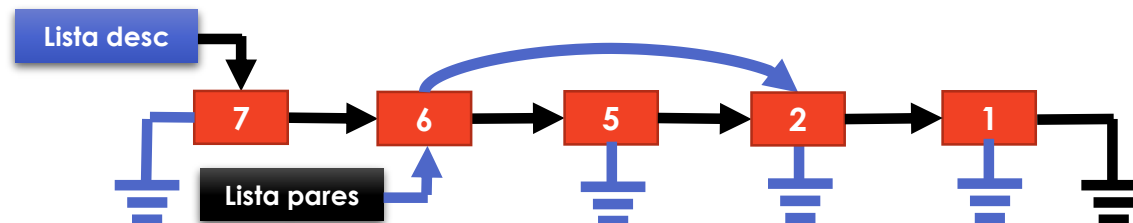
Estructuras de almacenamiento combinadas

Para representar el almacenamiento de la información es común utilizar estructuras de almacenamiento combinadas.

Las estructuras combinadas pueden generarse a partir de estructuras estáticas y/o dinámicas, y en distintas cantidades.



También puede darse la necesidad de que una determinada estructura proponga más de un criterio sobre sus datos, como por ejemplo una lista ordenada por un criterio y la misma lista ordenada por otro criterio.



Estructuras de almacenamiento combinadas

Ejemplo: arreglo de listas.

const

MAX=10;

type

punterolista=^nodo;

nodo=record

dato:integer;

ste:punterolista;

end;

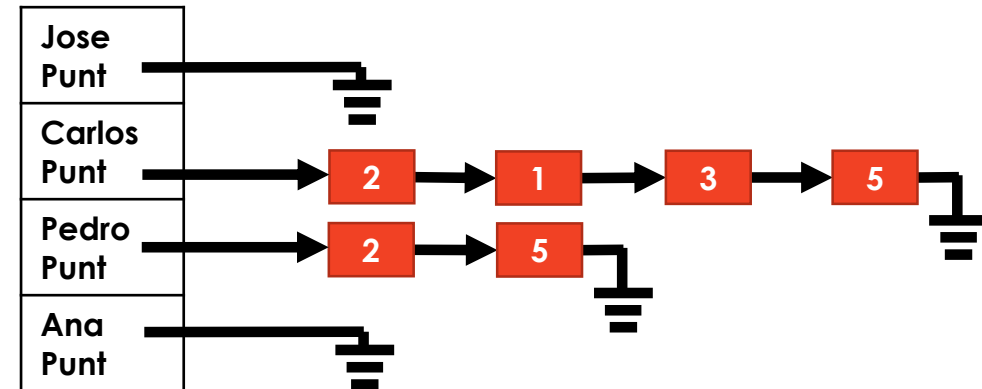
regarr=record

nombre:string;

punt:punterolista;

end;

arreglo=array [1..MAX] of regarr;



Observación: la estructura estática guarda información estática dentro de su tipo y la información dinámica a través de un puntero a dicha estructura.

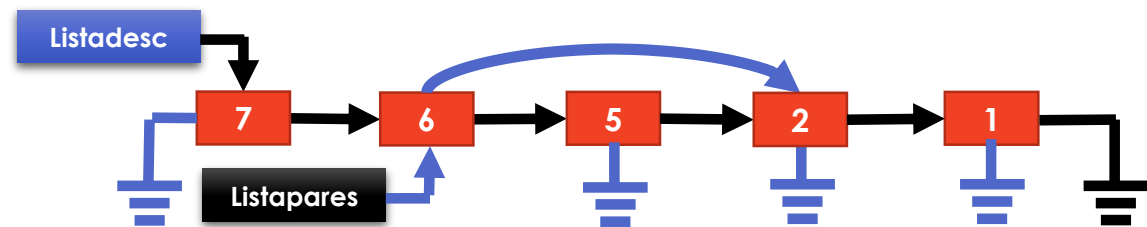
Estructuras de almacenamiento combinadas

Ejemplo: lista de nodos que mantiene más de un criterio entre sus elementos.

type

```
punterolista=^nodo;  
nodo=record  
    dato:integer;  
    steord:punterolista;  
    stepar:punterolista;  
end;
```

Var listadesc,listapares:punterolista;



Observación: la información o el dato es uno o más, y por cada criterio de orden mantiene un puntero del mismo tipo.

Estructuras de almacenamiento combinadas

La lista de índice se utiliza para hacer referencia a nodos de otra estructura para generar un nuevo criterio de orden entre los elementos.

type

```
punterolista=^nodo;
```

```
punterolistaind=^nodoindice;
```

```
nodo=record
```

```
    dato:integer;
```

```
    ste:punterolista;
```

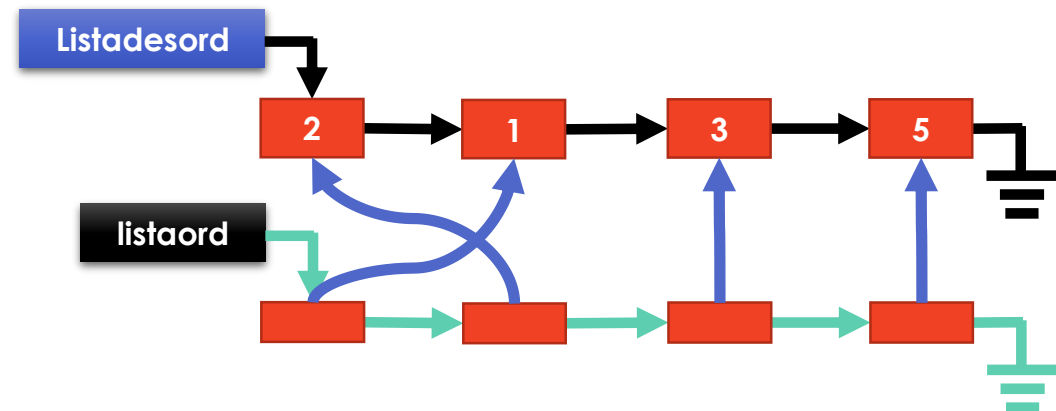
```
end;
```

```
nodoindice=record
```

```
    dato:punterolista;
```

```
    ste:punterolistaind;
```

```
end;
```



Observación: la información o el dato de lista de índices es una **indirección** a donde está la información real.

Estructuras de almacenamiento combinadas

En la lista de listas, cada nodo de la primera hace referencia a otra lista de nodos (del mismo tipo o no).

type

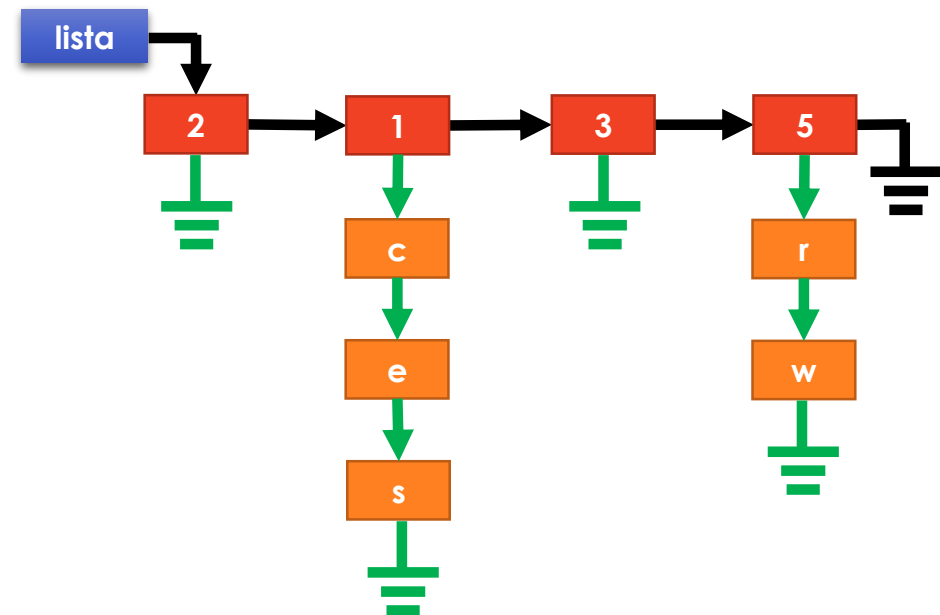
```
punterolistanum=^nodonum;  
punterolistachar=^nodochar;
```

```
nodonum=record  
    dato:integer;  
    ste:punterolistanum;  
    punt:punterolistachar;  
end;
```

end;

```
nodochar=record  
    dato:char;  
    ste:punterolistachar;  
end;
```

end;



Ejemplo: lista de clientes donde por cada cliente se mantiene las facturas.

Estructuras de almacenamiento combinadas

Supongamos que tenemos una lista que mantiene un criterio de orden, y además cada nodo tiene otro puntero con valor nil para armar otro criterio de orden.

//Definición de constantes y tipos

//Definición de métodos

Var listadesc,listapares,nodo:**punterolista**;

Begin

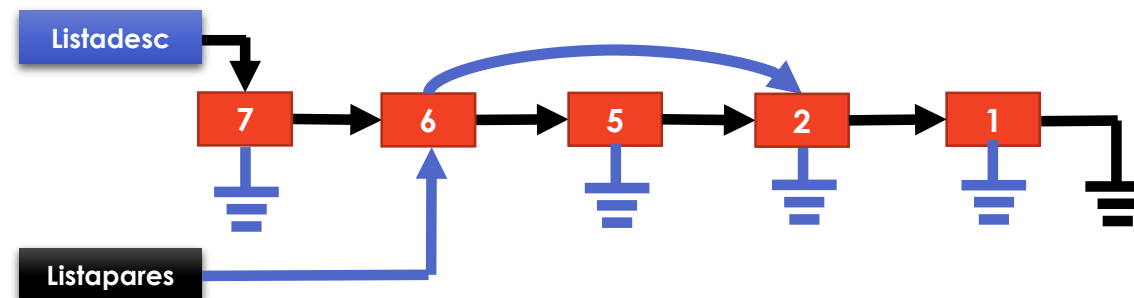
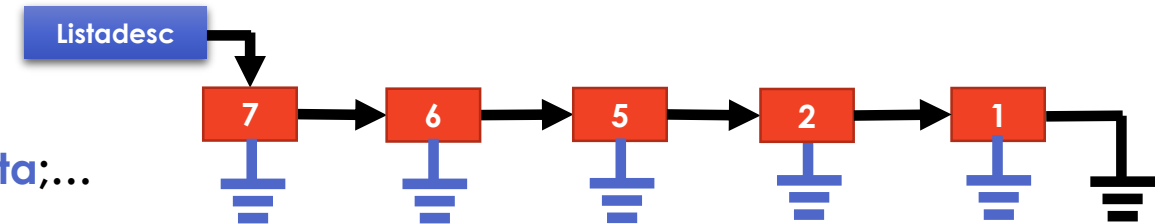
//Suponiendo que listadesc ya se encuentra creada

//Por cada nodo de la lista listadesc

if (nodo^.dato mod 2 = 0) **then**

insertarnodolistapares(listapares,nodo);

End.



Estructuras de almacenamiento combinadas

Procedure insertarnodolistadesc(**var listadesc:punterolista;nodo:punterolista**);

Begin

if (listadesc=NIL) or (listadesc^.dato<=nodo^.dato) then begin

nodo^.steord:=listadesc;

 listadesc:=nodo;

end

Else

 insertarnodolistadesc(listadesc^.**steord**,nodo);

End;

Procedure insertarnodolistapares(**var listapares:punterolista;nodo:punterolista**);

Begin

if (listapares=NIL) or (listapares^.dato<=nodo^.dato) then begin

nodo^.stepar:=listapares;

 listapares:=nodo;

end

Else

 insertarnodolistapares(listapares^.**stepar**,nodo);

End;

Estructuras de almacenamiento combinadas

//Definición de constantes y tipos

//Definición de métodos

Var listadesord,nodo:punterolista;...
 listaord,nodoindice:punterolistaindice;...

Begin

//Suponiendo que se necesita armar las dos listas por cada numero ingresado

Por cada numero que ingresa el usuario

begin

 crearnodo(nodo,numero);

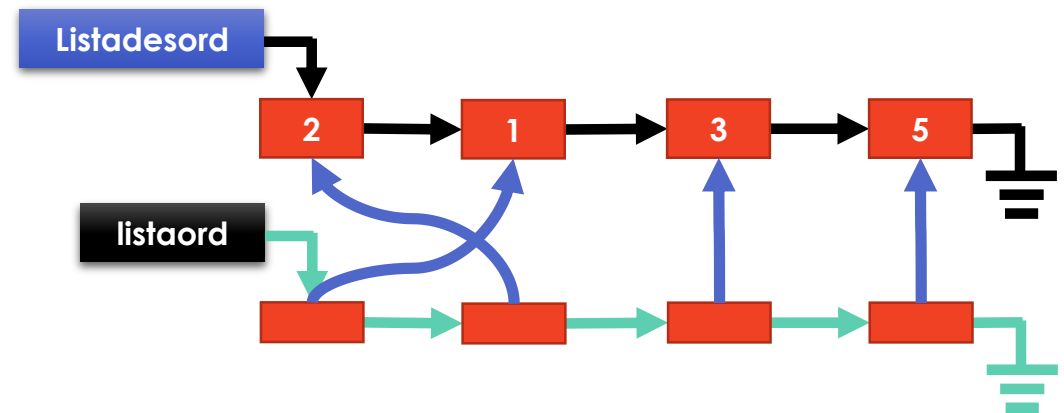
 agregarnodolistadesord(listadesord,nodo);

 crearnodoindice(nodoindice,nodo);

 insertarnodolistaind(listaord,nodoindice);

end;

End.



Estructuras de almacenamiento combinadas

Procedure agregarnodolistadesord(var listadesord:punterolista;nodo:punterolista);

Begin

 nodo^.ste:=listadesord;

 listadesord:=nodo;

End;

Procedure insertarnodolistaind(var listaord:punterolistaindice;nodoindice:punterolistaindice);

Begin

if (listaord=NIL) or (listaord^.dato^.dato<=nodoindice^.dato^.dato) **then begin**

 nodoindice^.ste:=listaord;

 listaord:=nodoindice;

end

Else

 insertarnodolistaind(listaord^.ste,nodoindice);

End;

Estructuras de almacenamiento combinadas

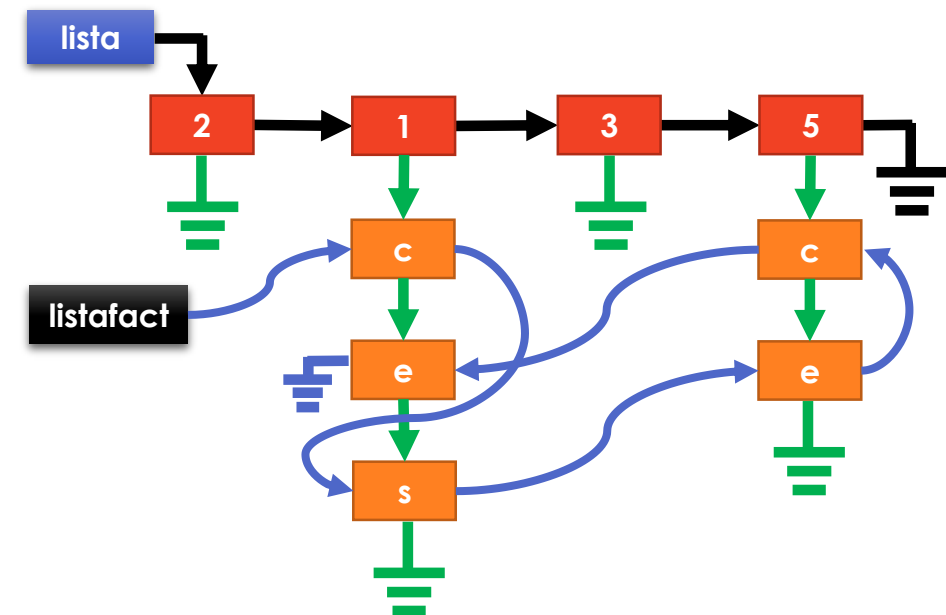
Ejemplo: lista de listas, donde cada nodo de la primera hace referencia a nodos de otra estructura, y se mantiene además un criterio de orden sobre todos los nodos de la segunda lista.

type

```
punterolistanum= $\wedge$ nodonum;  
punterolistachar= $\wedge$ nodochar;
```

```
nodonum=record  
  dato:integer;  
  ste:punterolistanum;  
  punt:punterolistachar;  
end;
```

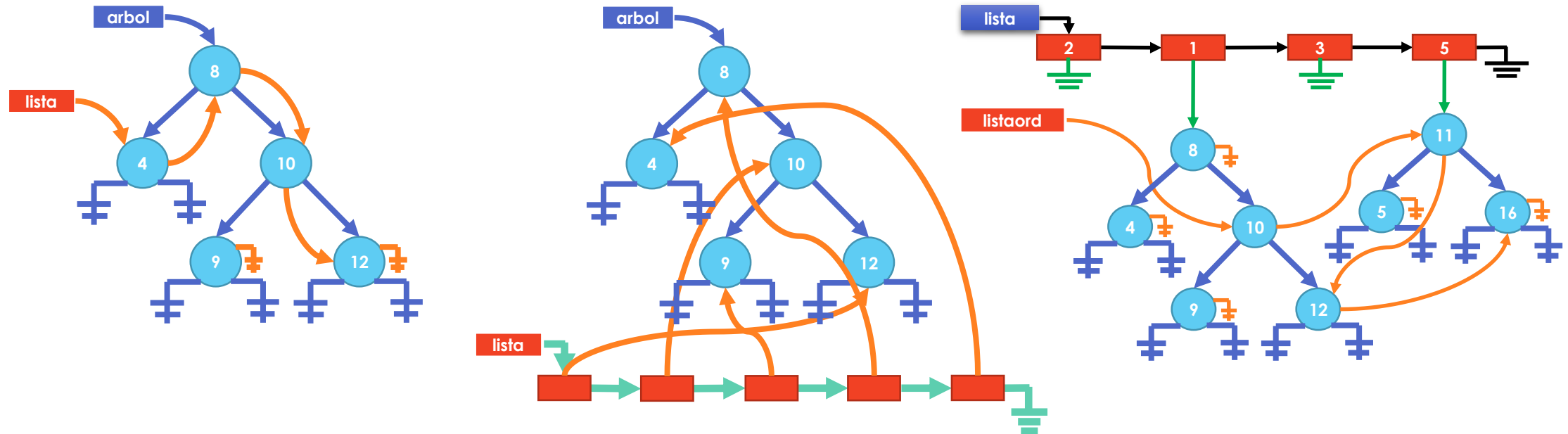
```
nodochar=record  
  dato:char;  
  ste:punterolistachar;  
  steord:punterolistachar;  
end;
```



Estructuras de almacenamiento combinadas

Cuando trabajamos con arboles, las combinaciones de estructuras se incrementa considerando que se puede mezclar arreglos, matrices, listas y arboles.

En este sentido las combinaciones más utilizadas en la materia consideran arboles y listas (y en cada uno de sus tipos: ordenadas, dobles, circulares, indirecciones o índices, etc.).



Estructuras de almacenamiento combinadas

Ejemplo: árbol ordenado y lista mantiene otro criterio de orden a través de un puntero extra de los nodos del árbol. El árbol que mantiene más de un criterio entre sus elementos.

type

```
punteroarbol=^nodoarbol;
```

```
nodoarbol=record
```

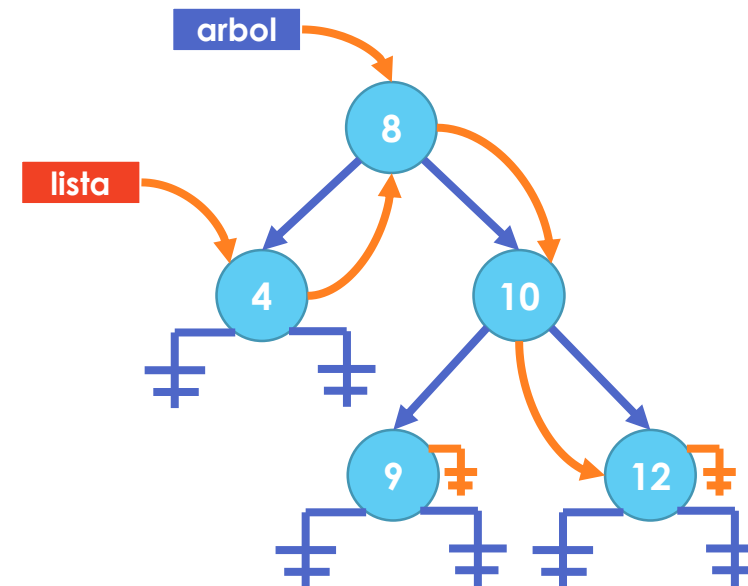
```
  dato:integer;
```

```
  izq,der,ste:punteroarbol;
```

```
end;
```

Var

```
  arbol,lista:punteroarbol;
```



Observación: la información o el dato es uno o más, con lo cual puede mantener muchos criterios de orden sobre un mismo tipo de nodo.

Estructuras de almacenamiento combinadas

Ejemplo: árbol ordenado y lista de índices al árbol. El árbol mantiene un criterio de orden entre sus elementos, y la lista de índices mantiene otro criterio.

type

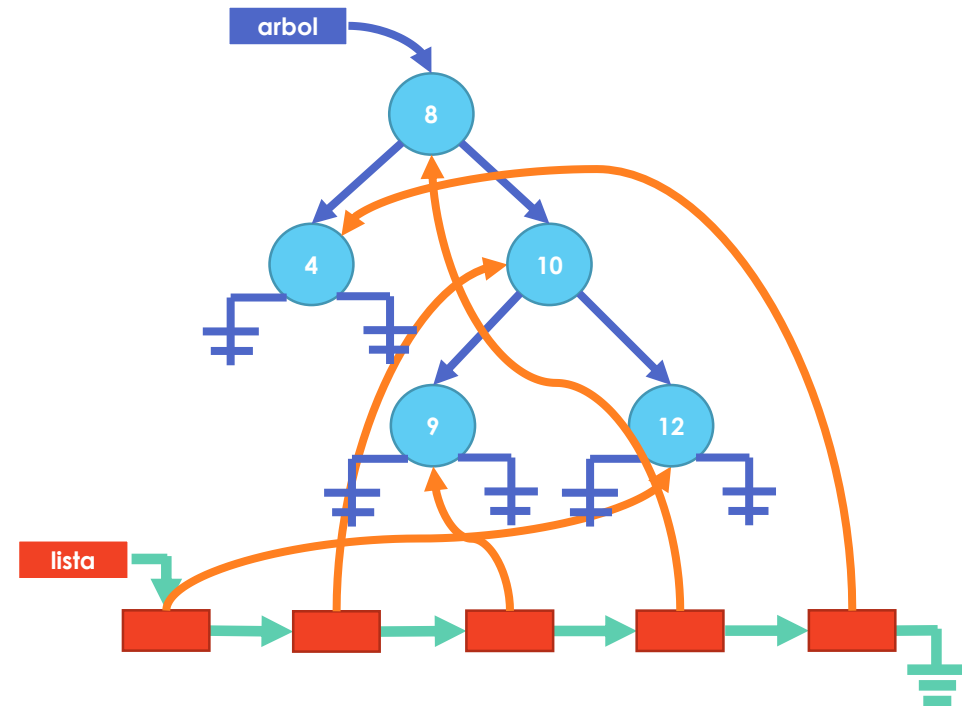
```
punteroarbol=^nodoarbol;  
punterolista=^nodolista;
```

```
nodoarbol=record  
    dato:integer;  
    izq,der:punteroarbol;
```

```
end;  
nodolista=record  
    dato:punteroarbol;  
    ste:punterolista;
```

```
end;
```

Var árbol:punteroarbol;
 lista:punterolista;



Observación: la información está en un solo lugar, en los nodos del árbol.

Estructuras de almacenamiento combinadas

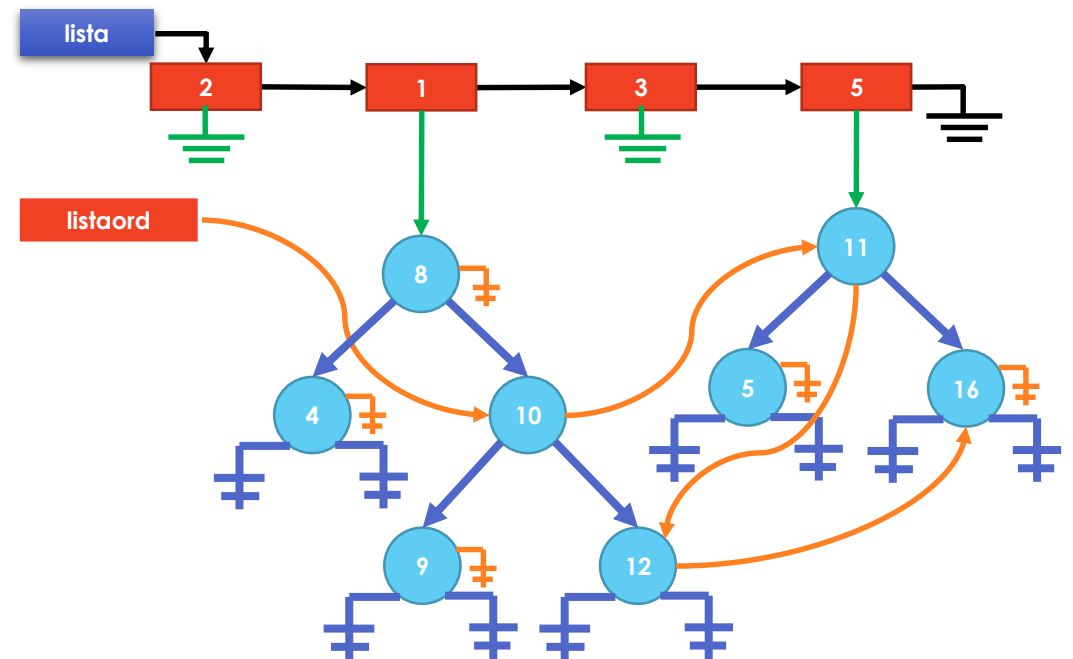
Ejemplo: lista de arboles y una lista extra que mantiene otro criterio de orden entre los nodos de los arboles.

type

```
punteroarbol=^nodoarbol;  
punterolista=^nodolista;  
nodolista=record  
    dato:integer;  
    ste:punterolista;  
end;  
nodoarbol=record  
    dato:integer;  
    izq,der,steord:punteroarbol;  
end;
```

Var

```
lista:punterolista;  
listaord:punteroarbol;
```



Observación: para llegar a un nodo del árbol primero me ubico en la lista, y dentro de esta recorro el árbol hasta llegar al nodo.

Estructuras de almacenamiento combinadas

Opción 1: crear el árbol y mientras lo hace crear la lista.

//Definición de constantes y tipos
//Definición de métodos

Var arbol,lista,nodo:parbol;...

Begin

//Suponiendo que ambas estructuras no están creadas

//Por cada numero que ingresa el usuario crear ambas estructuras

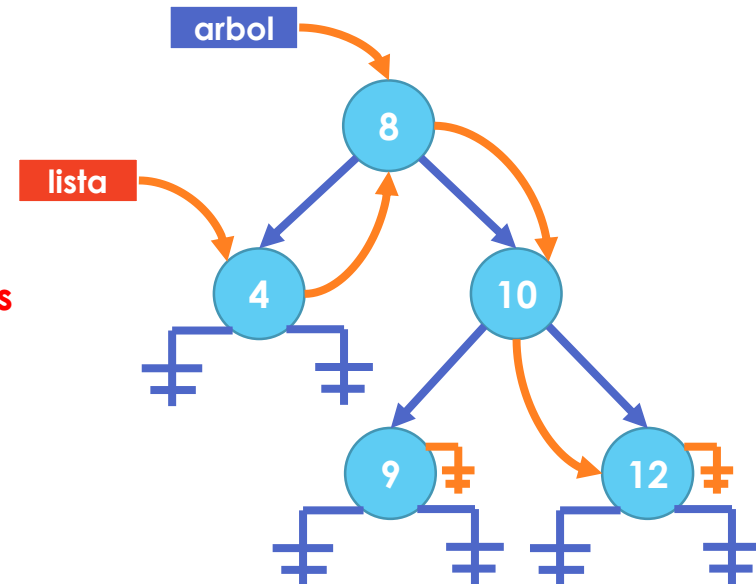
begin

crearnodo(nodo,numero);

insertarordenadoarbolylista(arbol,lista,nodo);

End;

End.



Estructuras de almacenamiento combinadas

```
Procedure insertarordenadoárbolylista(var árbol, lista:parbol; nodo:parbol);
```

```
Begin
```

```
if árbol=NIL then begin
```

```
    árbol:=nodo;
```

```
    insertarordenadolista(lista,nodo);
```

```
end
```

```
else if árbol^.dato>nodo^.dato then
```

```
    insertarordenadoárbolylista(árbol^.izq, lista, nodo)
```

```
else
```

```
    insertarordenadoárbolylista(árbol^.der, lista, nodo)
```

```
End;
```

```
Procedure insertarordenadolista(var lista:parbol;nodo:parbol);
```

```
Begin
```

```
if (lista=NIL) or (lista^.dato>=nodo^.dato) then begin
```

```
    nodo^.ste:=lista;
```

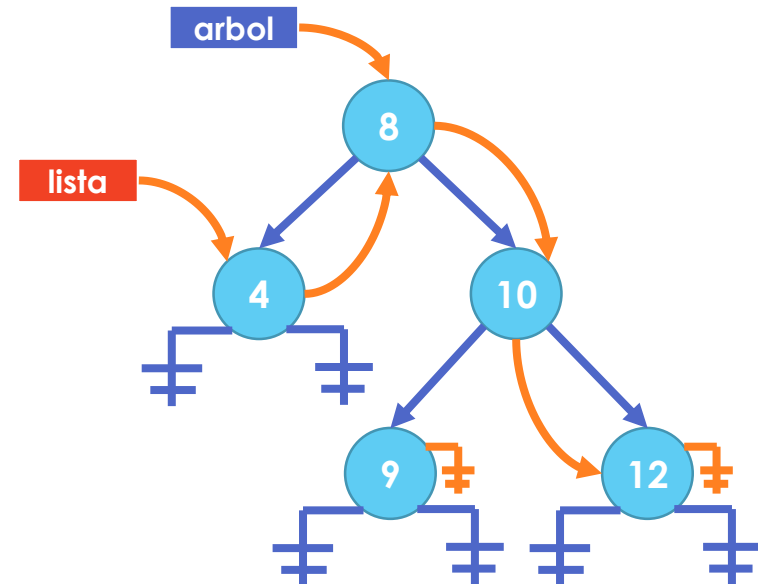
```
    lista:=nodo;
```

```
end
```

```
Else
```

```
    insertarordenadolista(lista^.ste, nodo);
```

```
End;
```



Estructuras de almacenamiento combinadas

Opción 2: crear el árbol y crear la lista se realiza por separado.

//Definición de constantes y tipos

//Definición de métodos

Var arbol,lista,nodo:parbol;...

Begin

//Suponiendo que ambas estructuras no están creadas

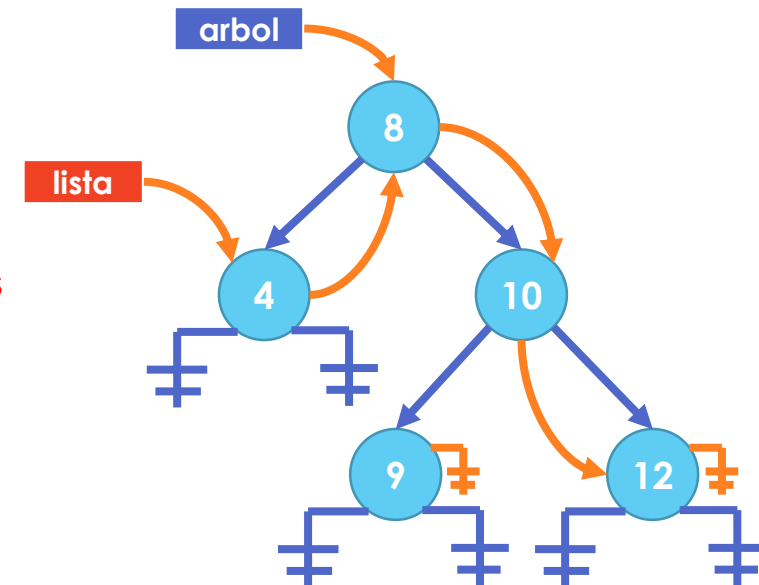
//Por cada numero que ingresa el usuario crear ambas estructuras

begin

```
crearnodo(nodo,numero);  
insertarordenadoarbol(árbol,nodo);  
insertarordenadolista(lista,nodo);
```

End;

End.



Estructuras de almacenamiento combinadas

```
Procedure insertarordenadoárbol(var árbol:parbol; nodo:parbol);
```

```
Begin
```

```
if árbol=NIL then begin
```

```
    árbol:=nodo;
```

```
end
```

```
else if árbol^.dato>nodo^.dato then
```

```
    insertarordenadoárbol(árbol^.izq,nodo)
```

```
else
```

```
    insertarordenadoárbol(árbol^.der,nodo)
```

```
End;
```

```
Procedure insertarordenadolista(var lista:parbol;nodo:parbol);
```

```
Begin
```

```
if (lista=NIL) or (lista^.dato>=nodo^.dato) then begin
```

```
    nodo^.ste:=lista;
```

```
    lista:=nodo;
```

```
end
```

```
Else
```

```
    insertarordenadolista(lista^.ste,nodo);
```

```
End;
```

