

ESTRUCTURAS DE DATOS

TIPOS ABSTRACTOS DE DATOS ARBORESCENTES

# Recorridos de árboles binarios

Manuel Montenegro Montes  
Departamento de Sistemas Informáticos y Computación  
Facultad de Informática – Universidad Complutense de Madrid

# ¿Qué es un recorrido?

Como recorrer una lista o de un array o de nodos, pero ahora con árboles.

- **Recorrer** un árbol significa visitar los nodos de un árbol, de modo que cada nodo es visitado exactamente una vez.  
Ningún nodo se quedará sin visitar y no hay ningún nodo que se visitará dos veces.
- **Visitar** un nodo significa realizar una acción específica, que puede depender del valor contenido dentro de ese nodo.  
Cosas que podemos hacer con los nodos:
  - Imprimir por pantalla el valor del nodo.
  - Sumar el valor del nodo a una variable externa.
  - Escribir el valor del nodo en un fichero.
  - Incrementar un contador externo.

**Comenzaremos aquí**

Nosotros vamos a ver 4 formas de recorrer los árboles binarios.

Nosotros en estos primeros vídeos vamos a considerar que visitar un nodo consiste en imprimir por pantalla su valor.

# Tipos de recorridos

Dos categorías:

Profundidad

Anchura.

1

- Recorrido en profundidad

*Depth First Search (DFS)*

- Preorden
- Inorden
- Postorden

Primero hijo izquierdo completamente

Después el hijo derecho completamente.

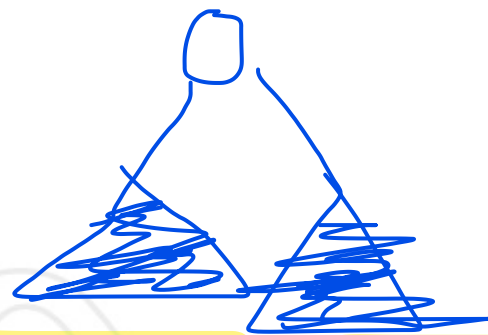
Depende de cuando visitemos la raíz.

2

- Recorrido en anchura

*Breadth First Search (BFS)*

Se recorre un poco del hijo izquierdo un poco del derecho... así sucesivamente.



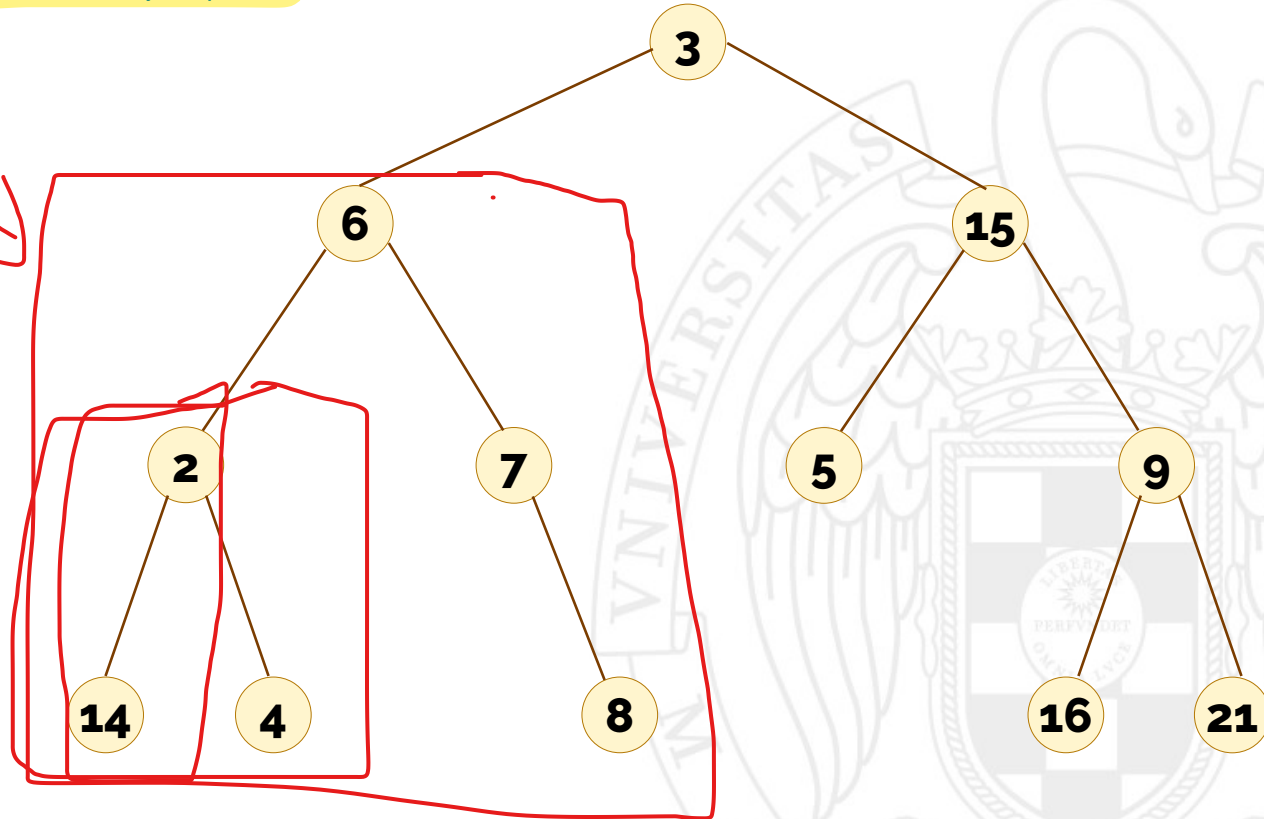
# Recorridos en profundidad

- Se explora completamente un hijo antes de pasar al siguiente.

Normalmente primero se visita el hijo izquierdo.

1º Izquierdo

En algunos casos primero es el hijo izquierdo, en otros casos a lo mejor primero visitamos la raíz.



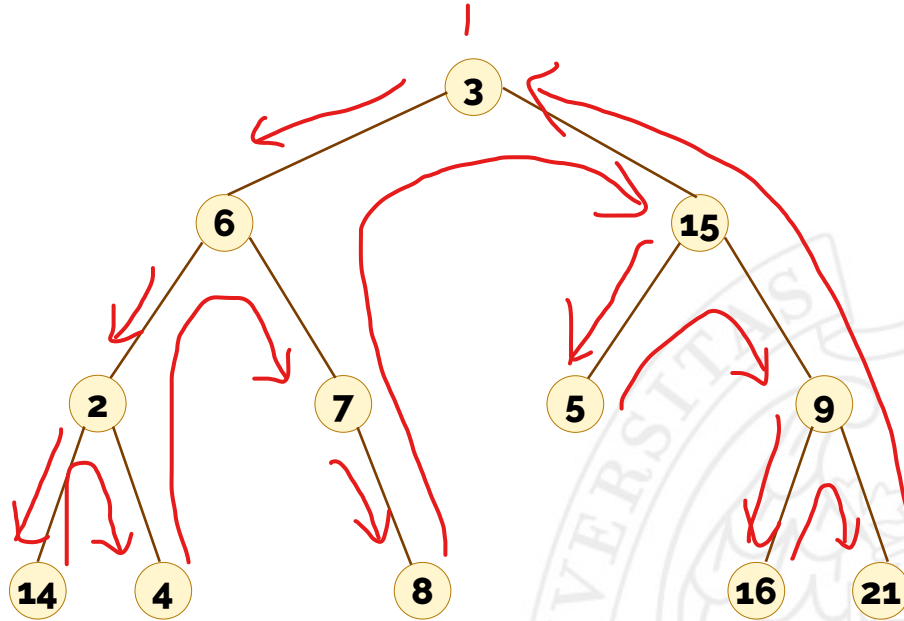
# Recorridos en profundidad

Vamos a ver los tres tipos de recorridos en profundidad.

- Se explora completamente un hijo antes de pasar al siguiente.
- **Preorden:** Visitar raíz, luego recorrer hijo izquierdo, luego recorrer hijo derecho.  
1º raíz, 2º hijo izquierdo y 3º hijo derecho



# Recorrido en preorden



Raíz del árbol

Hijo izquierdo

Hijo derecho

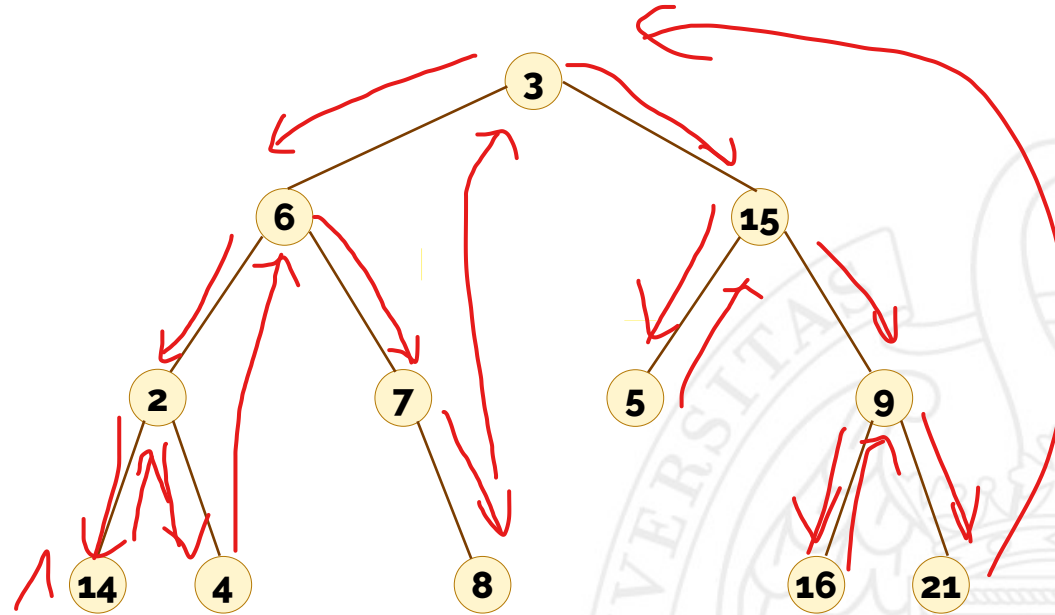


# Recorridos en profundidad

- Se explora completamente un hijo antes de pasar al siguiente.
  - **Preorden:** Visitar raíz, luego recorrer hijo izquierdo, luego recorrer hijo derecho.
  - **Inorden:** Recorrer hijo izquierdo, visitar raíz, luego recorrer hijo derecho.  
1º Hijo izquierdo, 2º raíz y 3º Hijo derecho



# Recorrido en inorden



14 2 4 6 7 8

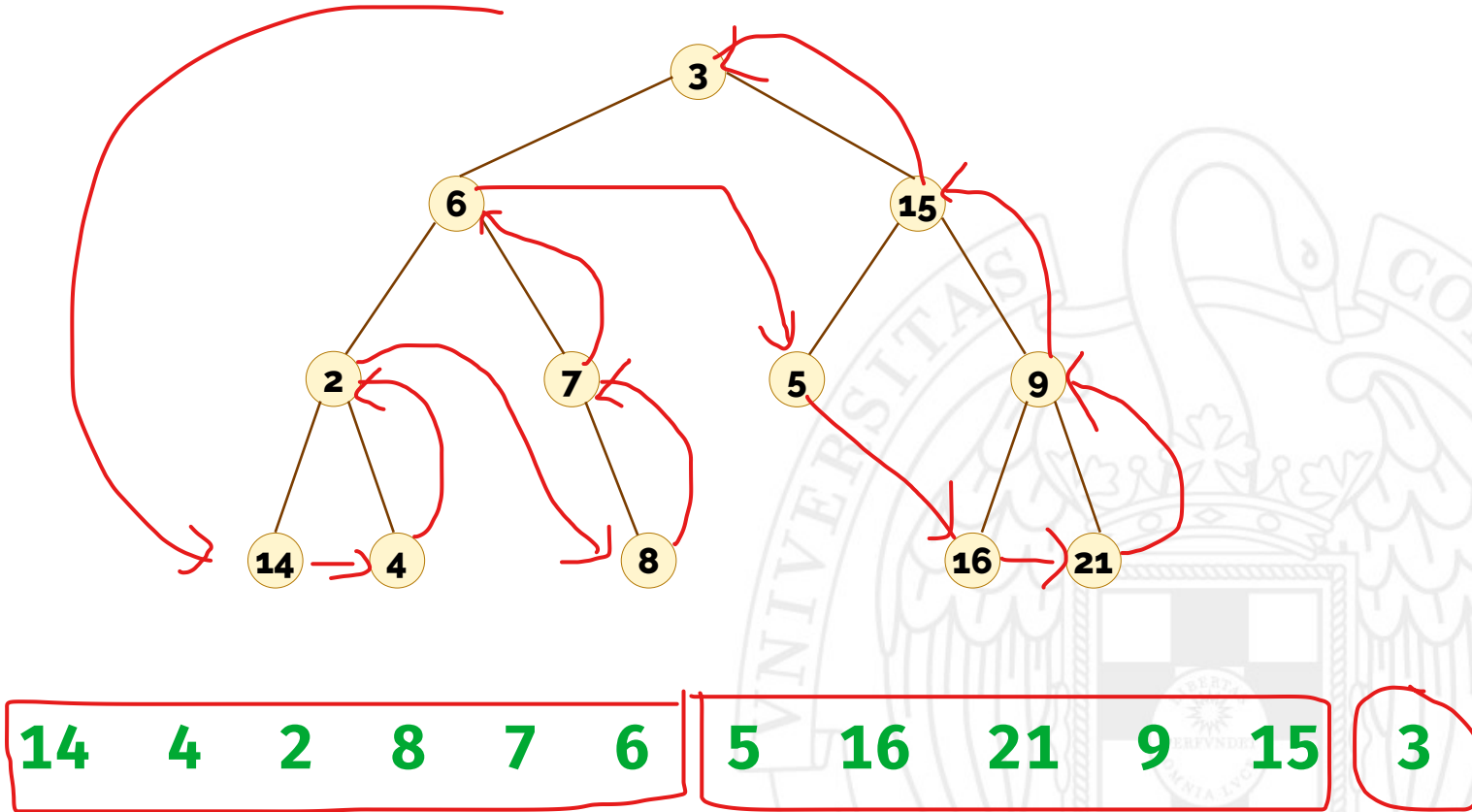
3 5 15 16 9 21



# Recorridos en profundidad

- Se explora completamente un hijo antes de pasar al siguiente.
  - **Preorden:** Visitar raíz, luego recorrer hijo izquierdo, luego recorrer hijo derecho.
  - **Inorden:** Recorrer hijo izquierdo, visitar raíz, luego recorrer hijo derecho.
  - **Postorden:** Recorrer hijo izquierdo, luego recorrer hijo derecho, luego visitar raíz. 1ºHijo izquierdo 2º Hijo derecho 3º raíz del árbol

# Recorrido en postorden



# Recorridos en profundidad

- Se explora completamente un hijo antes de pasar al siguiente.
  - **Preorden:** Visitar raíz, luego recorrer hijo izquierdo, luego recorrer hijo derecho.
  - **Inorden:** Recorrer hijo izquierdo, visitar raíz, luego recorrer hijo derecho.
  - **Postorden:** Recorrer hijo izquierdo, luego recorrer hijo derecho, luego visitar raíz.

# Tipos de recorridos

- Recorrido en profundidad  
*Depth First Search (DFS)*

- Preorden
- Inorden
- Postorden

- Recorrido en anchura  
*Breadth First Search (BFS)*

También se le llama recorridos por niveles.

# Recorridos en anchura (*por niveles*)

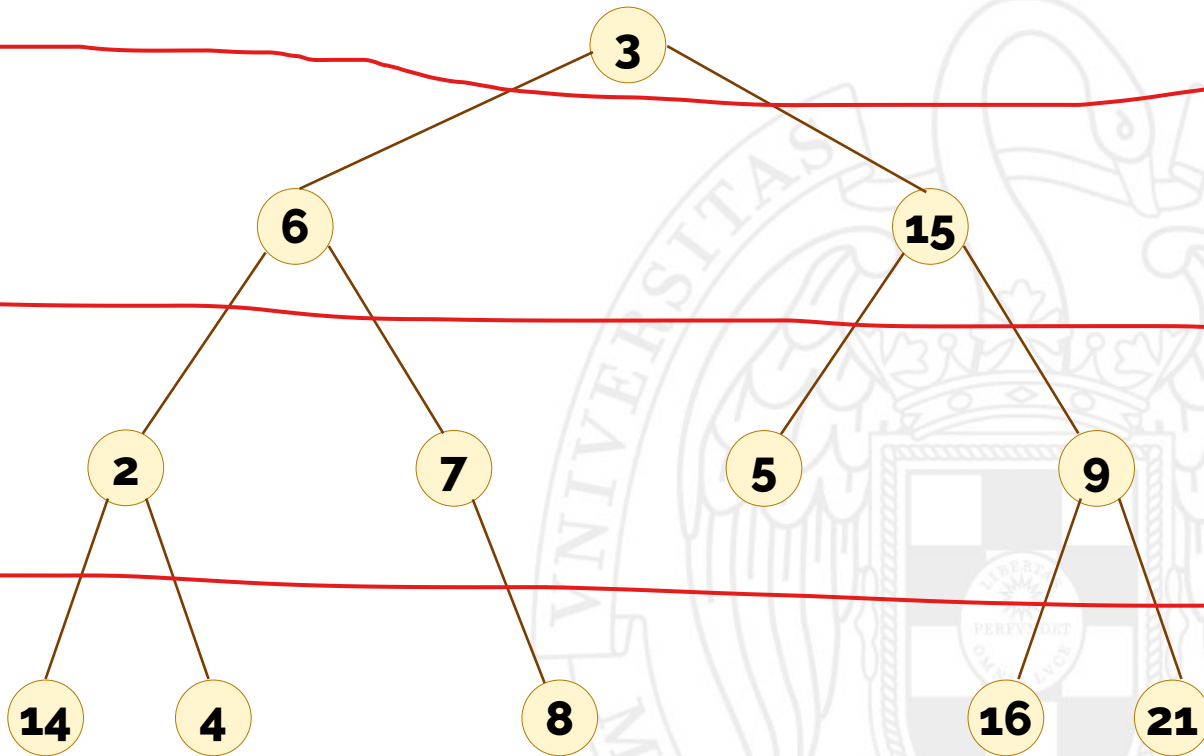
- Se explora completamente un nivel antes de pasar al siguiente.

Nivel 1

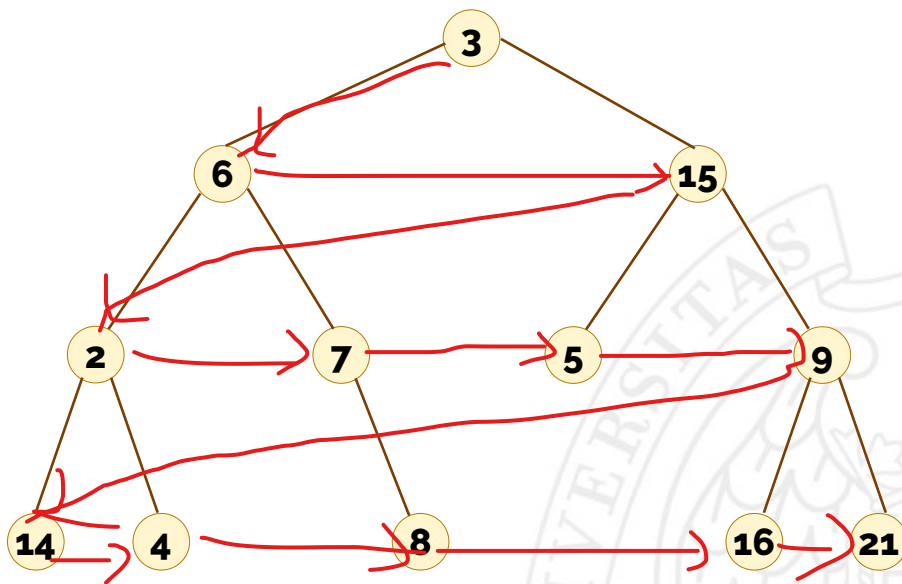
Nivel 2

Nivel 3

Nivel 4



# Recorrido en anchura (*por niveles*)



A través de la cola de prioridad.

3 6 15 2 7 5 9 14 4 8 16 21

Es el más difícil de implementar.