



UNIVERSIDAD  
DE MÁLAGA

Dpto. Lenguajes y  
Ciencias de la Computación

# Programación de Sistemas y Concurrencia Control 27/4/2012

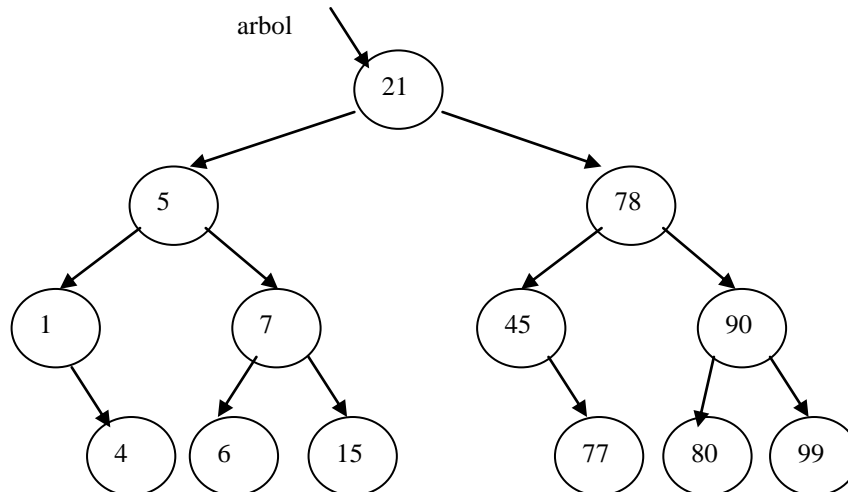
APELLIDOS \_\_\_\_\_ NOMBRE \_\_\_\_\_

DNI \_\_\_\_\_ ORDENADOR \_\_\_\_\_ GRUPO \_\_\_\_\_

1. Un árbol binario de búsqueda es un árbol binario en el que **para cualquier nodo** el subárbol izquierdo (si no está vacío) contiene valores menores que el valor que contiene dicho nodo, y el subárbol derecho (si no está vacío) contiene valores mayores. Así, por ejemplo, para la lista de números introducidos por este orden:

21, 5, 78, 1, 7, 45, 90, 4, 6, 15, 77, 80, 99

Se crearía un árbol binario de búsqueda con la siguiente distribución:



Definir el tipo TArbol e implementar las siguientes operaciones:

**void** CrearABB (TArbol \*arb)

Este procedimiento construye un árbol binario de búsqueda vacío.

**void** InsertarEnABB (TArbol \*arb, int elem)

Este procedimiento inserta elem en un árbol binario de búsqueda. Después de la inserción el árbol **DEBE** seguir siendo un árbol binario de búsqueda.

**void** RecorrerABB (TArbol arb)

Dado un árbol binario de búsqueda, este procedimiento muestra en pantalla los elementos del árbol ordenados de menor a mayor. Para el dibujo de la figura la salida sería:

1,4,5,6,7,15,21,45,77,78,80,90,99

**void** DestruirABB (TArbol \*arb)

Este procedimiento libera la memoria de todos los nodos del árbol.

2. Desarrollar un programa en Java que permita calcular de manera concurrente el máximo de un array de elementos de tipo `int`. Para calcular el máximo suponer que se dispone de un array de N posiciones y M hebras donde N y M podrán ser leídos desde teclado o almacenados como constantes. Los valores contenidos en el array son inicializados aleatoriamente antes de la creación de las hebras.

La solución desarrollada dividirá el array N en M fragmentos de manera que cada una de las hebras calculará el máximo del fragmento de array que le corresponda (se puede suponer para simplificar que N es múltiplo de M). Posteriormente, una vez hayan terminado todas las hebras de calcular su máximo local, en la hebra principal del programa se procederá a calcular el máximo global examinando el máximo local de cada una de las hebras.

La siguiente figura ilustra el proceso para un array de 9 posiciones y 3 hebras.

