

TÉCNICAS DE PROGRAMACIÓN AVANZADAS

INTRODUCCIÓN AL DISEÑO DE ALGORITMOS

ACTIVIDAD APLICATIVA INDIVIDUAL

Ejercicio 1. Escribir en Java una función que reciba como entrada un árbol binario y que devuelva como resultado cuántos nodos tiene.

Ejemplo de posible cabecera:

```
int contarNodos (ArbolBin a)
```

Ejercicio 2. Escribir en Java una función que reciba como entrada un árbol binario y que devuelva como su profundidad (el número de niveles que tiene).

Ejemplo de posible cabecera:

```
int profundidad (ArbolBin a)
```

Ejercicio 3. Codificar en Java una función que devuelva verdadero o falso si un árbol binario dado está lleno o no teniendo en cuenta su actual profundidad. Calcular y razonar la complejidad de la solución propuesta.

Ejemplo de posible cabecera:

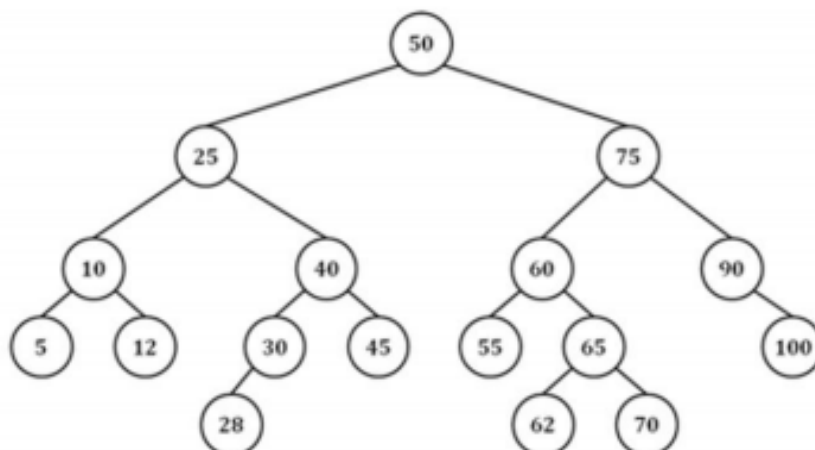
```
boolean esLleno (ArbolBin a)
```

Ejercicio 4. A partir del siguiente árbol AVL, se pide resolver las siguientes operaciones:

- Insertar el 61 a partir del árbol de la figura.
- Eliminar el 45 a partir del árbol de la figura.
- Insertar el 29 a partir del árbol de la figura.

Para cada una de ellas se pide:

- Explicar si se produce o no un desequilibrio.
- En caso afirmativo, identificar el nodo desequilibrado.
- Razonar qué rotación o rotaciones habría que aplicar.
- Dibujar paso a paso cada una de ellas.



TÉCNICAS DE PROGRAMACIÓN AVANZADAS
INTRODUCCIÓN AL DISEÑO DE ALGORITMOS
ACTIVIDAD APLICATIVA INDIVIDUAL

Ejercicio 5. Codificar en Java una función que reciba como entrada un árbol binario y determine, devolviendo verdadero o falso, si dicho árbol es un árbol binario de búsqueda.