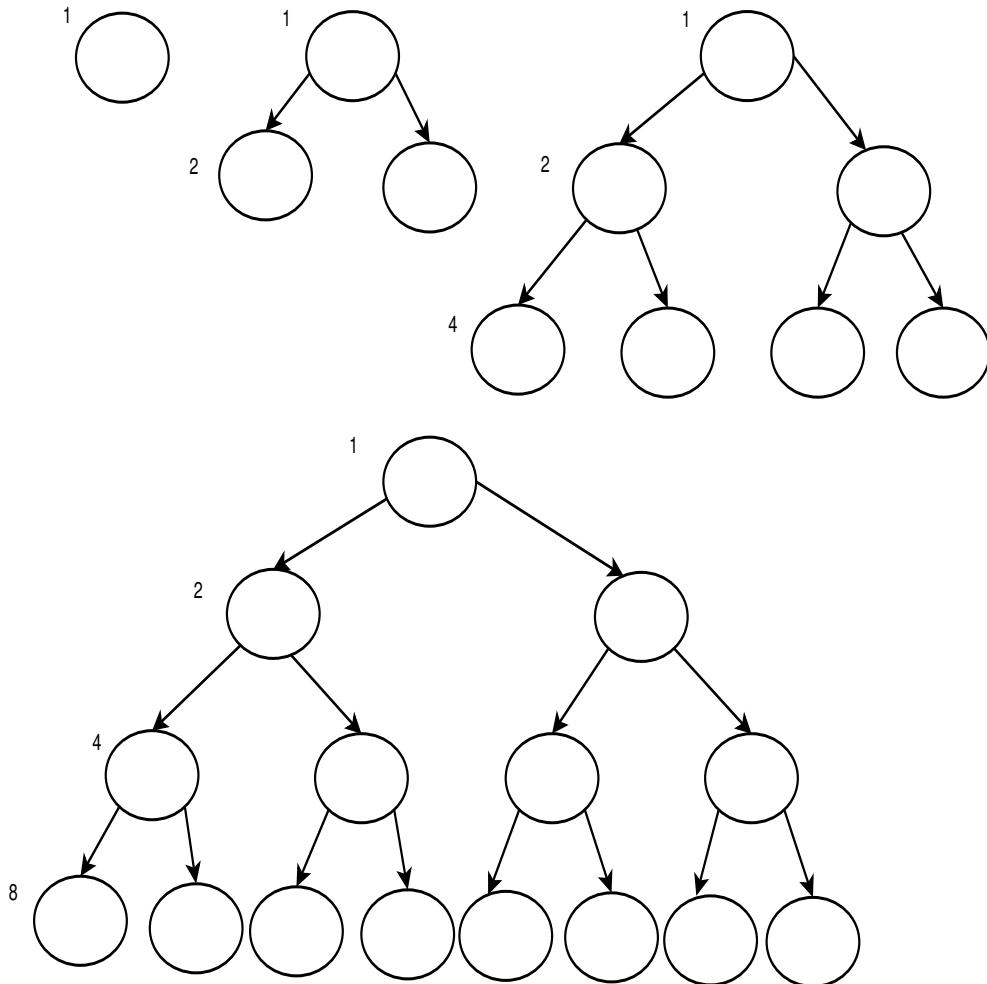


# Propiedades de Árboles Binarios

**Nivel de profundidad  $d$ .** Es el conjunto de nodos que tienen la misma profundidad  $d$  en el árbol binario.

**Árbol binario lleno.** Sea  $T$  un árbol binario con profundidad  $d$ . Decimos que  $T$  está lleno si todos los nodos que están en un nivel menor a  $d$  tienen dos nodos hijos.

El árbol binario lleno es el que tiene la mayor cantidad de nodos posible de todos los árboles binarios con esa misma profundidad.



Recordemos que podemos usar altura y profundidad indistintamente para estas definiciones. Veremos a continuación algunas propiedades importantes de los árboles binarios llenos.

**Teorema 1.** Un árbol binario lleno  $T$  con profundidad  $d$  tiene exactamente  $2^i$  nodos en el nivel  $i$ , donde  $0 \leq i \leq d$ .

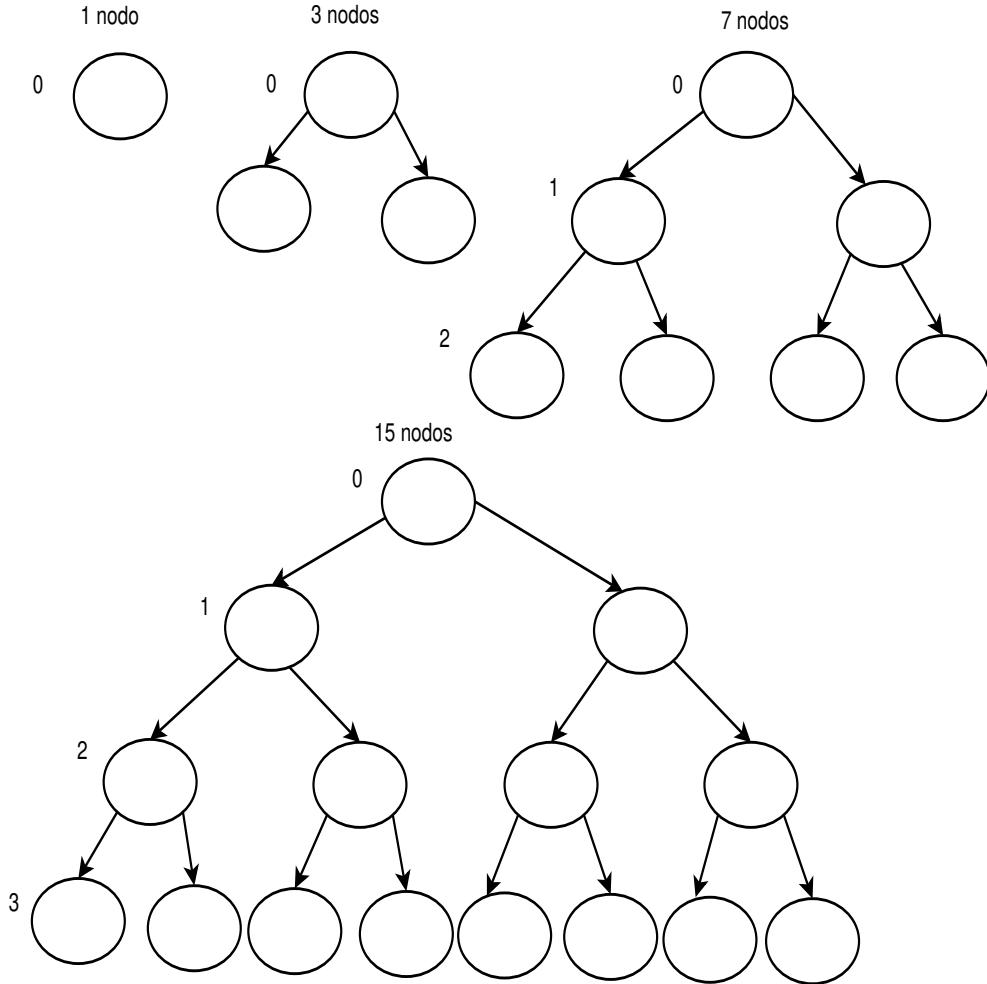
Dem. Por inducción sobre  $i$ .

Caso base:  $i = 0$ .  $T$  consiste de un solo nodo que es hoja y por tanto  $T$  tiene  $1 = 2^0$  nodos.

H.I. Supongamos que el teorema es válido para los niveles  $0 \leq i < d$ .

Paso inductivo. Consideremos el nivel  $i = d$  de profundidad, el cual consiste sólo de hojas. En particular, por H.I. tenemos que el nivel  $d - 1$  tiene  $2^{d-1}$  nodos. Como el árbol es lleno, cada uno de los nodos del nivel  $d - 1$  tiene 2 hijos y por tanto el número de nodos en el nivel  $d$  es el doble de los del nivel  $d - 1$ , es decir, hay  $2 * 2^{d-1} = 2^d$  nodos.

Consecuencia. El número de hojas de un árbol binario lleno con profundidad  $d$  es  $2^d$ .



**Teorema 2.** Un árbol binario lleno  $T$  de profundidad  $d$  tiene  $2^{d+1} - 1$  nodos.

Dem. Por inducción sobre  $d$ .

Caso base:  $d = 0$ .  $T$  consiste de un solo nodo, pero  $1 = 2 - 1 = 2^{0+1} - 1$ .

H.I. Supongamos que el teorema es válido para  $d = n$ .

Paso inductivo. Consideremos un árbol binario lleno  $T$  de profundidad  $n + 1$ .

Como  $n + 1 > 0$  entonces  $T$  consiste de una raíz y dos subárboles de profundidad  $n$ , los cuales por H.I. tienen  $2^{n+1} - 1$  nodos cada uno. Por tanto,  $T$  tiene la suma de los nodos de ambos subárboles más uno, que es la raíz, es decir,  $T$  tiene  $2(2^{n+1} - 1) + 1 = 2^{(n+1)+1} - 1$

nodos, lo cual demuestra el teorema.

Como consecuencia, el número de nodos internos de un árbol binario lleno es el número de nodos total menos el numero de hojas:  $2^{d+1} - 1 - 2^d = 2 * 2^d - 2^d - 1 = 2^d - 1$ .

**Teorema 3.** La profundidad  $d$  de un árbol binario con  $n$  nodos satisface las siguientes desigualdades  $\log_2(n + 1) - 1 \leq d \leq n - 1$ .

Un árbol binario con profundidad  $d$  debe tener un número de nodos  $n$  que no exceda el que puede tener un árbol binario lleno con esa profundidad. El Teorema 2 establece que un árbol binario lleno con profundidad  $d$  tiene  $2^{d+1} - 1$  nodos, por lo que

$$n \leq 2^{d+1} - 1$$

$$\Rightarrow n + 1 \leq 2^{d+1}$$

$$\Rightarrow \log_2(n + 1) = \log_2 2^{d+1} = d + 1$$

$$\therefore d = \log_2(n + 1) - 1$$

Por otro lado, es posible construir un árbol binario con  $n$  nodos que forme una trayectoria que pase por todos los nodos desde la raíz hasta la única hoja, dicha trayectoria tiene longitud es  $n - 1$ , por lo que  $d \leq n - 1$ .

Ejemplo: La profundidad  $d$  de un árbol binario con 15 nodos es al menos  $\log_2 16 - 1 = 3$  y a lo más 14.

