Recorrido Preorden: Tiempo de Ejecución (Desarrollo)

Considerando un árbol binario lleno y altura h

$$T(n) = \begin{cases} c & n = 1 \\ c + 2\,T((n-1)/2) & n > 1 \end{cases}$$

$$\begin{aligned}
T(n) &= c + 2T((n-1)/2) && (1)\\
&= c + 2c + 2^2\,T((n-1-2)/2^2) && (2)\\
&= c + 2c + 2^2 c + 2^3\,T((n-1-2-2^2)/2^3) && (3)\\
&\;\;.\\
&\;\;.\\
&\;\;.\\
&= (2^i - 1)c + 2^i\,T((n-(2^i-1))/2^i) && (i)
\end{aligned}$$

$(n-(2^i-1))/2^i = 1 \Rightarrow n-(2^i-1) = 2^i \Rightarrow n+1 = 2^i + 2^i \Rightarrow n+1 = 2^{i+1} \Rightarrow$
$log_2(n+1) = i+1 \Rightarrow i = log_2(n+1) - 1$

$$\begin{aligned}
T(n) &= (2^{log_2(n+1)-1} - 1)c + 2^{log_2(n+1)-1}\,T((n-(2^{log_2(n+1)-1}-1))/2^{log_2(n+1)-1})\\
&= ((2^{log_2(n+1)}/2) - 1)c + (2^{log_2(n+1)}/2)\,T((n-((2^{log_2(n+1)}/2)-1))/2^{log_2(n+1)}/2)\\
&= (((n+1)/2) - 1)c + ((n+1)/2)\,T((n-(((n+1)/2)-1))/(n+1)/2)\\
&= (((n+1)/2) - 1)c + ((n+1)/2)\,T((n-(n+1-2)/2)/(n+1)/2)\\
&= (((n+1)/2) - 1)c + ((n+1)/2)\,T((n-(n-1/2))/(n+1)/2)\\
&= (((n+1)/2) - 1)c + ((n+1)/2)\,T((2n - n+1/2)/(n+1)/2)\\
&= (((n+1)/2) - 1)c + ((n+1)/2)\,T((n+1/2)/(n+1)/2)\\
&= (((n+1)/2) - 1)c + ((n+1)/2)\,T(1)\\
&= (((n+1)/2) - 1)c + ((n+1)/2)c = (n+1-1)c = nc \qquad \therefore O(n)
\end{aligned}$$

Otra forma: expresando en función de la altura

$$T(h) = \begin{cases} c & h = 0 \\ c + 2\,T(h-1) & h > 0 \end{cases}$$

$$
\begin{aligned}
T(h) &= c + 2T(h-1) && (1)\\
&= c + 2c + 2^2\,T(h-2) && (2)\\
&= c + 2c + 2^2 c + 2^3\,T(h-3) && (3)\\
&\;.\\
&\;.\\
&= (2^i - 1)c + 2^i\,T(h-i) && (i)
\end{aligned}
$$

$h-i = 0 \Rightarrow i=h$

$$
\begin{aligned}
T(h) &= (2^h - 1)c + 2^h\,T(h-h)\\
&= (2^h - 1)c + 2^h\,T(0)\\
&= (2^h - 1)c + 2^h c\\
&= (2^{h+1} - 1)c \quad \therefore O(n)
\end{aligned}
$$

En un árbol binario lleno de altura h la cantidad de nodos es 2^{h+1} -1