NOMBRE: Vicente Espinosa

SECCIÓN: 2

N° LISTA: 36

PUNTAJE:

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC1253 — Matemáticas Discretas — 1' 2020

Tarea 7 – Respuesta Pregunta 2

1

Sabemos que se cumple para 0 y para 1, pues $2^0 = 1$ y $2^1 = 2$, luego asumiendo que se cumple para $n-1$, lo demostraremos para n:

Dado que #nodes$(T_{n-1}) = 2^{n-1}$

Sabemos que #nodes(T_n) será $1 + \sum_{i=1}^{n-1}$#nodes(T_i)

por lo tanto, se puede traducir a $1 + \sum_{i=1}^{n-1} 2^i$

Y, siguiendo esta lógica, si juntamos el 1 que sobra, con 2^0, nos dará 2^1, el cual podemos sumar con el 2^1 que hay, y conseguir 2^2. Si seguimos así, deberíamos llegar a tener solo la suma de $2^{n-1} + 2^{n-1}$, lo que da como resultado final: 2^n.

2

Sabemos que depth(t) $= 1 + \{\text{depth}(t_1), ..., \text{depth}(t_k)\}$

Sabemos que depth$(T_0) = 0$

Luego, podemos calcular depth(T_1) usando lo anterior: depth(T_1) = 1 + máx{ depth(T_0) } = 1+0 = 1.

Luego, siguiendo lo mismo, depth(T_2) = 2 y asi sucesivamente, por lo tanto, notamos que depth(T_m) aumenta junto con m, entonces, si asumimos que depth(T_{n-1}) = n-1, debemos probar que depth(T_{n-1}) = n.

depth(T_{n-1}) = 1 + máx{ depth(T_{n-1}),..,depth(T_1) } = 1 + n - 1 = n.

Por lo tanto, para cada T_n, probamos que depth(T_n) = n

3

Nada

4

Puesto que ya probamos que #nodes$(T_i) = 2^i$, solo basta invertir esto:

#nodes$(T_i) = 2^i$ Aplicamos $\log_2$

$\log_2$(# nodes(T_i)) $= \log_2(2^i)$

$\log_2$(# nodes(T_i)) $= i$