Laboratorio 4 | Métodos para el análisis de recursión

Geordie Quiroa

August 28, 2018

1 – Método: Sustitución

– T(n) = T(n–1) + n

Supongo que la solución es: **T(n)** = $\mathbf{O(n^2)}$
T(n) = $\mathbf{c(n^2)}$

Asumo que la condición (m < n) se mantiene.
m = $\mathbf{(n-1)}$; parte del T(n) original

T(n) = $\mathbf{c(n-1)^2+n}$
T(n) = $\mathbf{c(n^2-2n+1)+n}$
T(n) = $\mathbf{cn^2-2cn+1c+n}$
T(n) = $\mathbf{cn^2-2cn+1c+n}$
T(n) = $\mathbf{cn^2-cn+1c}$
T(n) <= $\mathbf{cn^2}$
$\mathbf{O(n^2)}$ **<=** $\mathbf{cn^2}$

Caso base **n = 2,3 y c = 2**

Sustituyo
T(n) <= $\mathbf{c(n)^2}$; $c\ =\ 2$
$\mathbf{O(n^2)}$ **<=** $\mathbf{c(n)^2}$

T(2) <= $\mathbf{2*(2)^2}$
4 <= $\mathbf{8}$

T(3) <= $\mathbf{2*(3)^2}$
9 <= $\mathbf{18}$
En ambos casos cumple con la condición de comparación. Por lo que la complejidad es $\mathbf{O(n^2)}$

– T(n) = T(n/2) + 1

Supongo que la solución es: **T(n)** = $\mathbf{O(lg(n))}$
T(n) = $\mathbf{c(log(n))}$

Asumo que la condición (m < n) se mantiene.
m = $\mathbf{(n/2)}$; parte del T(n) original
T(n) = $\mathbf{c(log(n/2))+1}$
T(n) = $\mathbf{c(log(n)\ -\ log(2))+1}$
T(n) = $\mathbf{c*log(n)+1}$
T(n) = $\mathbf{c*log(n)}$

T(n) <= $\mathbf{c * \log(n)}$
$\mathbf{O(\log(n))}$ <= $\mathbf{c * \log(n)}$

Caso base **n = 2,4 y c = 2**

Sustituyo
T(n) <= $\mathbf{c(\log(n))}$; $c = 2$
$\mathbf{O(\log(n))}$ <= $\mathbf{c(\log(n))}$

T(2) <= $\mathbf{2 * (\log(2))}$
1 <= 2

T(4) <= $\mathbf{2 * (\log(4))}$
2 <= $\mathbf{4}$
En ambos casos cumple con la condición de comparación. Por lo que la complejidad es $\mathbf{O(\log(n))}$

2 - Método: Árbol de Recursión

- T(n) = 3T(n/2) + n

0.0.1 Imágen del árbol de recursión adjunta en el repositorio.

Compruebo el resultado del árbol de recursión con el método de sustitución
Por medio del árbol de recursión, inferí que la solución de T(n) = 3T(n/2) + n es: **T(n) =** $\mathbf{O(n)}$
T(n) = $\mathbf{c(n)}$

Asumo que la condición (m < n) se mantiene.
m = $\mathbf{(n/2)}$; parte del T(n) original

T(n) = $\mathbf{3T(n/2) + n}$
T(n) = $\mathbf{3 * c * (n/2) + n}$
T(n) = $\mathbf{5/2 * c * n}$
T(n) = $\mathbf{c * n}$
T(n) <= $\mathbf{c * n}$
$\mathbf{O(n)}$ **<=** $\mathbf{c * n}$

Caso base **n = 2,3 y c = 2**

Sustituyo
T(n) <= $\mathbf{c(n)}$; $c = 2$

$\mathbf{O(n)}$ <= $\mathbf{c(n)}$

T(2) <= $\mathbf{2*(2)}$
2 <= $\mathbf{4}$

T(3) <= $\mathbf{2*(3)}$
3 <= $\mathbf{6}$
En ambos casos cumple con la condición de comparación. Por lo que la complejidad es $\mathbf{O(n)}$

3 - Método: Teorema Maestro | T(n) = aT(n/b) + f(n)

- T(n) = 2T(n/4) + 1

Evaluación

f(n) = 1
a = 2
b = 4
$\log_4 2$ = 1/2
ϵ = 1/2
CASO f(n) = $\mathbf{O(n^{\log_4 2-\epsilon})}$
$\mathbf{(1)}$ = $\mathbf{O(n^{1/2-\epsilon})}$

Debido a que f(n) varía por epsilon, se aplica el caso descrito en las líneas de arriba.
Por lo tanto la complejidad es T(n) = $\mathbf{\Theta(n^{\log_4 2})}$

- T(n) = 2T(n/4) + $\sqrt{n}$

Evaluación

f(n) = $\sqrt{\mathbf{n}}$
a = 2
b = 4
$\log_4 2$ = 1/2

CASO f(n) = $\mathbf{O(n^{\log_4 2})}$
$\mathbf{(n^{1/2})}$ = $\mathbf{O(n^{1/2})}$

Por lo tanto la complejidad es T(n) = $\mathbf{\Theta(n^{\log_4 2}\log_2(n))}$ = $\mathbf{\Theta(n^{1/2}\log_2(n))}$

– T(n) = 2T(n/4) + n

Evaluación

f(n) = n
a = 2
b = 4
$\log_4 \mathbf{2}$ = 1/2
ϵ = 1/2
CASO f(n) = $\mathbf{O}(\mathbf{n}^{\log_4 \mathbf{2}+\epsilon})$
$\mathbf{O}(\mathbf{n})$ = $\mathbf{O}(\mathbf{n}^{\mathbf{1/2}-\epsilon})$
Para validar este caso, hace falta evaluar la siguiente condición
a*f(n/b): <= c*f(n) ; c<1
c = 1/2
$(2/4)(n^2)$ <= $(1/2)(n)$

Debido a que f(n) varía por epsilon, y la condición se validó, la complejidad es T(n) = $\Theta(\mathbf{n})$.

– T(n) = 2T(n/4) + n^2

Evaluación

f(n) = $\mathbf{n^2}$
a = 2
b = 4
$\log_4 \mathbf{2}$ = 1/2
ϵ = 3/2
CASO f(n) = $\mathbf{O}(\mathbf{n}^{\log_4 \mathbf{2}+\epsilon})$
$\mathbf{O}(\mathbf{n})$ = $\mathbf{O}(\mathbf{n}^{\mathbf{1/2}-\epsilon})$
Para validar este caso, hace falta evaluar la siguiente condición
a*f(n/b): <= c*f(n) ; c<1
c = 1/2
$(2/4)(n^2)$ <= $(1/2)(n^2)$

Debido a que f(n) varía por epsilon, y la condición se validó, la complejidad es T(n) = $\Theta(\mathbf{n^2})$.