

Análisis de algoritmos recursivos

Caballero Hernandez Juan Daniel

Tarea 2:

No. Cuenta: 31729503-4

Ejercicios:

A) $x(n) = x(n-1) + 5$ para $n > 1$, $x(1) = 0$

Procedimiento:

Al sustituir $x(n-1) = x(n-2) + 5$

$$x(n) = [x(n-2) + 5] + 5$$

$$= x(n-2) + 10 \quad \text{Al sustituir } x(n-2)$$

$$= x(n-3) + 5$$

$$= [x(n-3) + 5] + 10$$

$$= x(n-3) + 15$$

Resulta en

$$x(n) = x(n-i) + 5i$$

Tenemos que la condición inicial $x(1) = 0$ y sí $i = n-1$

$$x(n) = x(n-(n-1)) + 5n$$

$$= x(1) + 5n$$

$$= 0 + 5n$$

$$= 5n \rightarrow n$$

Es de orden lineal = $x(n)$

B) $x(n) = 3x(n-1)$ para $n > 1$, $x(1) = 4$

Procedimiento:

Al sustituir $x(n-1) = 3x(n-2)$

$$x(n) = 3[3x(n-2)]$$

$$= 9x(n-2) \quad \text{Al sustituir } x(n-2) = 3x(n-3)$$

$$= 9[3x(n-3)]$$

$$= 27x(n-3)$$

$$x(n) = 3^i x(n-i)$$

Tenemos que la condición inicial $x(1) = 4$ y si $i=n-1$

$$x(n) = 3^n x(n-(n-1))$$

$$x(n) = 3^n x(1)$$

$$x(n) = 3^n (4)$$

$$x(n) = 3^n$$

Ese de orden exponencial = 3^n

C) $x(n) = x(n-1) + n$ para $n > 0$, $x(0) = 0$

Procedimiento:

Al sustituir $x(n-1) = x(n-2) + n$

$$x(n) = [x(n-2)+n]+n$$

$$= x(n-2) + 2n$$

$$x(n) = x(n-i) + in$$

Tenemos que la condición inicial $x(0) = 0$ y si $i=n$

$$x(n) = x(0) + n^2$$

$$x(n) = n^2$$

Ese de orden cuadrático = n^2

D) $x(n) = x(n/2) + n$ para $n > 1$, $x(1)=1$ (Resolver para $n = 2^k$)

Sustituir $x(n/2) = x(n/2/2) + n/2 = x(n/4) + n/2$

$$x(n) = [x(n/4)+n/2] +n$$

$$= x(n/4) + 3n/2 \quad \text{Al sustituir } x(n/4) = x(n/8) + n/4$$

$$= [x(n/8)+n/4] + 3n/2$$

$$= x(n/8) + 7n/4$$

$$x(n) = x(n/2^i) + n[(2^i)-1] / \{(2^i)-[2^{(i-1)}]\}$$

Tenemos que la condición inicial $x(1)=1$, $n=2^k$ y si $i=\log_2(n)$

Algoritmo_misterioso(n):

```
s<- 0  
Para i <- 1 hasta n hacer  
    s<-s + i*i  
Fin Para  
Devolver s  
Fin
```

¿Qué calcula el algoritmo?

La suma de los cuadrados de los números que van desde 0 hasta n

¿Cuál es la operación básica?

La suma

¿Cuántas veces se ejecuta la operación básica?

n veces

¿Cuál es la eficiencia del algoritmo?

$M(n)$ de Orden lineal

Sugerir una mejora al algoritmo.

Algoritmo_misterioso(n):

```
Si n < 2 entonces  
    Devolver 1  
Fin si  
Devolver ( $n * n$ ) + (Algoritmo_misterioso( $n-1$ ) * Algoritmo_misterioso( $n-1$ ))
```