Actividad 3: Notación de orden.

Diego Javier Mora Cortés

7 de Noviembre 2022

CUCEI - INGENIERÍA INFORMÁTICA

Algoritmia

Profesor: David Alejandro Gómez Anaya

Índice

1. Afirmaciones de orden

¿Cuáles de las siguientes afirmaciones son verdaderas y cuales son falsas? Demuestre sus respuestas.

1.1. $n^2 \in O(n^3)$

$$n^2 \leq cn^3$$

$$1 \leq \frac{cn^3}{n^2}$$

$$1 \leq cn$$

Verdadero, para cualquier c>0 y $n_0 = 1$

1.2. $n^2 \in \Omega(n^3)$

$$n^2 \geq cn^3$$

$$1 \geq \frac{cn^3}{n^2}$$

$$1 \geq cn$$

$$\frac{1}{c} \geq n$$

$$\text{vemos que: } \max \frac{1}{c} = 1$$

$$1 \geq n$$

Falso, para cualquier $c \geq 1, n_0 = 2$

1.3. $2^n \in \Theta(2^{n+1})$

$$2^n \in \mathrm{O}(2^{n+1})$$

$$2^n \leq c2^{n+1}$$

$$\frac{2^n}{2^n} \leq \frac{c2^{n+1}}{2^n}$$

$$1 \leq 2c$$

Verdadero para cualquier $c \geq \frac{1}{2}$ y $n_0 = 1$

$$2^n \in \Omega(2^{n+1})$$

$$2^n \geq c2^{n+1}$$

$$\frac{2^n}{2^n} \geq \frac{c2^{n+1}}{2^n}$$

$$1 \geq 2c$$

Verdadero para cualquier $c \leq \frac{1}{2}$ y $n_0 = 1$

1.4. $n! \in \Theta(n+1)!$

$$n! \in \Omega((n+1)!)$$

$$n! \geq c(n+1)!$$

$$\frac{n!}{n!} \geq \frac{c(n+1)!}{n!}$$

$$1 \geq c(n+1)$$

$$1 \geq cn + c$$

$$\lim_{n\to\infty} 1 \geq \lim_{n\to\infty} cn + c$$

$$1 \geq \infty + c$$

Falso para cualquier $c \in \mathbb{R}^+$

2. Desarrollar algoritmos partiendo de una complejidad

Para las siguientes notaciones de orden, dicte algoritmos que con que su complejidad pertenezca a las notaciones. Calcule la complejidad de sus algoritmos y demuestre la pertenencia a las notaciones.

2.1. O(n)

Entrada: $a[n], q, s \in \mathbb{N}$
Salida: imprime: $a[i]$ | q y s son factores

Algorithm 1 Factores en arreglo

```
1: for i ← 0, i < n , i++ do
2:     if a[i] mod q == 0 and a[i] mod s == 0 then
3:         imprime: a[i]
4:     end if
5: end for
```

Complejidad

$$T(n) = 2 + \sum_{i=0}^{n-1}(10)$$

$$T(n) = 2 + (n - 1 - 0 + 1)(10)$$

$$T(n) = 2 + (n)(10)$$

$$\boldsymbol{T(n) = 10n + 2}$$

$$\boldsymbol{T(n) \in O(n)}$$

2.2. $\Omega(1)$

Entrada: $a, b \in \mathbb{N}$
Salida: $a + b$

Algorithm 2 Suma dos

```
1: resultado ← a + b
2: return resultado
```

Complejidad

$$\boldsymbol{T(n) = 3}$$

$$\boldsymbol{T(n) \in \Omega(1)}$$

2.3. $\Theta(\log_2(n))$

Entrada: $n \in \mathbb{N}$
Salida: p | potencia de 2 menor mas cercana a n

Algorithm 3 Potencia cercana

```
1: p ← 1
2: cont ← 0
3: while p < n do
4:     p ← 2 · p
5:     cont + +
6: end while
7: return cont
```

Complejidad

$$T(n) = 4 + \sum_{1}^{\log_2(n)} (4)$$

$$T(n) = 4 + (\log_2(n) - 1 + 1)(4)$$

$$T(n) = 4 + 4 \cdot \log_2(n)$$

$$\boldsymbol{T(n) = 4 \cdot \log_2(n) + 4}$$

$$\boldsymbol{T(n) \in \Theta(\log(n))}$$

3. Evaluar algoritmos según complejidad

De los siguientes algoritmos, analice su complejidad temporal, determine y demuestre las notaciones O grande, Omega mayúscula y Theta que definen la complejidad de cada algoritmo.

3.1. Algoritmo 1

Algorithm 4 Primer algoritmo

```
1: for i ← 0, i < n , i++ do
2:     for j ← 0, j < n , j++ do
3:         b ← A[j][i]
4:         A[j][i] ← A[i][j]
5:         A[i][j] ← b
6:     end for
7: end for
```

Complejidad

$$\boldsymbol{T_1(n)}\textbf{ : líneas 2-6}$$

$$T_1(n) = 2 + \sum_{j=0}^{n-1}(11)$$

$$T_1(n) = 2 + (n - 1 - 0 + 1)(11)$$

$$T_1(n) = 2 + 11 \cdot n$$

$$T_1(n) = 11n + 2$$

$$\boldsymbol{T(n) = 2 + \sum_{i=0}^{n-1}(T_1(n))}$$

$$T(n) = 2 + (n - 1 - 0 + 1)(T_1(n))$$

$$T(n) = 2 + (n)(T_1(n))$$

$$T(n) = 11n^2 + 2n + 2$$

Comparaciones

$$\boldsymbol{T(n) \in \Omega(n)}$$

$$11n^2 + 2n + 2 \geq cn \textbf{ para } n_0 = 1, c = 1$$

$$\boldsymbol{T(n) \in \Theta(n^2)}$$

$$11n^2 + 2n + 2 \in \Omega(n^2)$$

$$11n^2 + 2n + 2 \geq cn^2 \textbf{ para } n_0 = 1, c = 1$$

$$11n^2 + 2n + 2 \in \mathrm{O}(n^2)$$

$$11n^2 + 2n + 2 \leq cn^2 \textbf{ para } n_0 = 1, c = 15$$

$$\boldsymbol{T(n) \in \mathrm{O}(n^3)}$$

$$11n^2 + 2n + 2 \leq cn^3 \textbf{ para } n_0 = 1, c = 15$$

3.2. Algoritmo 2

Algorithm 5 Segundo algoritmo

```
1: for i ← 1, i ≤ n ,i++ do
2:     for j ← 1, j ≤ i ,j++ do
3:         sum++
4:     end for
5:     for k ← 1, k ≤ n ,k++ do
6:         A[k] = k − 1
7:     end for
8: end for
```

Complejidad

$T_1(n)$: líneas 2-4

$$T_1(n) = 2 + \sum_{j=1}^{i}(3)$$

$$T_1(n) = 2 + (3)(i - 1 + 1)$$

$$T_1(n) = 3 \cdot i + 2$$

$T_2(n)$: líneas 5-7

$$T_2(n) = 2 + \sum_{k=1}^{n}(5)$$

$$T_2(n) = 2 + (n - 1 + 1)(5)$$

$$T_2(n) = 5n + 2$$

$$\boldsymbol{T(n) = 2 + \sum_{i=1}^{n}(T_1(n) + T_2(n))}$$

$$T(n) = \sum_{i=1}^{n}((3i + 2) + (5n + 2))$$

$$T(n) = \sum_{i=1}^{n}(5n + 3i + 4)$$

$$T(n) = \sum_{i=1}^{n}(5n + 4) + \sum_{i=1}^{n}(3i)$$

$$T(n) = n(5n + 4) + \frac{3n(n + 1)}{2}$$

$$T(n) = 5n^2 + 4n + \frac{3n^2 + 3n}{2}$$

$$T(n) = \frac{10n^2 + 8n + 3n^2 + 3n}{2}$$

$$T(n) = \frac{13n^2 + 11n}{2}$$

Comparaciones

$$\boldsymbol{T(n) \in \Omega(1)}$$

$$\frac{13n^2 + 11n}{2} \geq cn \textbf{ para } n_0 = 1, c = 1$$

$$\boldsymbol{T(n) \in \Theta(n^2)}$$

$$\frac{13n^2 + 11n}{2} \in \Omega(n^2)$$

$$\frac{13n^2 + 11n}{2} \geq cn^2 \textbf{ para } n_0 = 1, c = 1$$

$$\frac{13n^2 + 11n}{2} \in \mathrm{O}(n^2)$$

$$\frac{13n^2 + 11n}{2} \leq cn^2 \textbf{ para } n_0 = 1, c = 12$$

$$\text{e.g: } \frac{13 + 11}{2} \leq 12$$

$$\text{e.g: } 12 \leq 12$$

$$\boldsymbol{T(n) \in \mathrm{O}(n^3)}$$

$$\frac{13n^2 + 11n}{2} \leq cn^3 \textbf{ para } n_0 = 1, c = 12$$

3.3. Algoritmo 3

Algorithm 6 Tercer algoritmo

```
1: for i ← 0, i < n ,i++ do
2:     c ← i
3:     while  c > 1 do
4:         x ← a + b
5:         c ← c/2
6:     end while
7: end for
```

Complejidad

$T_1(n)$: líneas 3-6

$$T_1(n) = 1 + \sum_{1}^{\log_2(i)} (5)$$

$$T_1(n) = 5\lfloor \log_2(i) \rfloor + 1$$

$$\boldsymbol{T(n) = 2 + \sum_{i=1}^{n} (3 + T_1(n))}$$

$$T(n) = 2 + \sum_{i=1}^{n} (5\log_2(i) + 4)$$

$$T(n) = 2 + 5 \cdot \sum_{i=1}^{n} (\log_2(i)) + \sum_{i=1}^{n} (4)$$

$$T(n) = 2 + 5 \cdot \sum_{i=1}^{n} (\log_2(i)) + \sum_{i=1}^{n} (4)$$

$$T(n) = 2 + 5 \cdot log_2(n!) + 4n$$

Dada la aproximacion de Stirling's

$$\log(n!) = n\log(n) - n + \Theta(\log n)$$

tenemos:

$$T(n) = 2 + 5 \cdot (n\log(n) - n + \log n + 4n$$

$$T(n) = 2 + 5n\log(n) - 5n + 5\log n + 4n$$

$$T(n) = 5n\log(n) + 5\log n - n + 2$$

Comparaciones

$$\boldsymbol{T(n) \in \Omega(n)}$$

$$5n\log(n) + 5\log n - n + 2 \geq cn \textbf{ para } n_0 = 1, c = 1$$

$$\boldsymbol{T(n) \in \Theta(n\log(n))}$$

$$5n\log(n) + 5\log n - n + 2 \in \Omega(n\log(n))$$

$$5n\log(n) + 5\log n - n + 2 \geq n\log(n) \textbf{ para } n_0 = 1, c = 1$$

$$5n\log(n) + 5\log n - n + 2 \in \mathrm{O}(n^2)$$

$$5n\log(n) + 5\log n - n + 2 \leq cn^2 \textbf{ para } n_0 = 1, c = 12$$

$$\boldsymbol{T(n) \in \mathrm{O}(n^2)}$$

$$5n\log(n) + 5\log n - n + 2 \leq cn^3 \textbf{ para } n_0 = 2, c = 10$$