Universidad Simón Bolívar
Departamento de Computación y
Tecnología de la Información
Estructuras Discretas para el Análisis de Algoritmos
Ci-2525k.

Solución Prática 6

1. Determine la dominación asintótica si existe, por O grande y Ω, entre las funciones dadas a continuación:

   a. $f_1 : \mathbb{N} \mapsto \mathbb{R}$ tal que $f_1(n) = n^2$

   b. $f_2 : \mathbb{N} \mapsto \mathbb{R}$ tal que $f_2(n) = n^2 + 1000n$

   c. $f_3 : \mathbb{N} \mapsto \mathbb{R}$ tal que

   $$f_3(n) \begin{cases} n & \text{si } n \text{ es par} \\ n^3 & \text{si } n \text{ es impar} \end{cases}$$

   d. $f_4 : \mathbb{N} \mapsto \mathbb{R}$ tal que

   $$f_4(n) \begin{cases} n & \text{si } n < 100 \\ n^3 & \text{si } n \geq 100 \end{cases}$$

   e. $f_5 : \mathbb{N} \mapsto \mathbb{R}$ tal que $f_5(n) = ln(n^{ln(2n)})$

   Solución: Recordemos la definición de O y Ω:

   O **grande:** $f(x)$ es $O(g(x))$ si y sólo si existe x_0 y una constante k positiva tal que $|f(x)| \leq k|g(x)|$ para todo $x \geq x_0$
   Ω **grande:** $f(x)$ es $O(g(x))$ si y sólo si existe x_0 y una constante k positiva tal que $|f(x)| \geq k|g(x)|$ para todo $x \geq x_0$

   Observación: en lo que sigue k será una constante positiva.
   Comparación de f_1 con f_2: f_1 es $O(f_2)$ ya que:
   $|f_1(n)| \leq k|f_2(n)| \iff |n^2| \leq k|n^2 + 1000n|$ se cumple para $k = 1$ y $n \geq 0$

   f_1 es $\Omega(f_2)$ ya que:

   $$\begin{aligned} |f_2(n)| &= |n^2 + 1000n| \\ &\leq |n^2 + 1000n^2| \\ &\leq |1001n^2| \\ &= 1001|n^2| \end{aligned}$$

   $\Rightarrow |f_2(n)| \leq 1001|f_1(n)| \iff |f_1(n)| \geq \frac{1}{1001}|f_2(n)|$, con $k = \frac{1}{1001}$ y $n \geq 0$

Observe que f_2 es $O(f_1)$ y también $\Omega(f_1)$, por lo tanto f_2 es $\Theta(f_1)$

Comparación de f_2 **con** f_1: f_2 es $O(f_1)$ ya que:

$$\begin{aligned} |f_2(n)| &= |n^2 + 1000n| \\ &\leq |n^2 + 1000n^2| \\ &\leq |1001n^2| \\ &= 1001|n^2| \end{aligned}$$

$\Rightarrow |f_2(n)| \leq 1001|f_1(n)|$, con $k = 1001$ y $n \geq 0$

f_2 es $\Omega(f_1)$ ya que: $|f_2(x)| \geq k|f_1(x)| \iff |n^2 + 1000n| \geq k|n^2|$, se cumple con $k = 1$ y $n \geq 0$

Observe que f_2 es $O(f_1)$ y también $\Omega(f_1)$, por lo tanto f_2 es $\Theta(f_1)$

Comparación de f_1 **con** f_3: f_1 **no** es $O(f_3)$ ya que aunque se cumple que $|n^2| \leq k|n^3|$, con $k = 1$ para todo $n \geq 0$ en el caso en que n es impar, para el caso en que n es par no se cumple que $|f_1| \leq k|f_3|$, por lo tanto f_1 **no** es $O(f_3)$. Un caso similar ocurre si comparanos f_3 con f_1.

f_1 **no** es $\Omega(f_3)$ ya que aunque se cumple que $|n^2| \geq k|n|$, con $k = 1$ para todo $n \geq 0$ en el caso en que n es par, para el caso en que n es impar no se cumple que $|f_1| \geq k|f_3|$, por lo tanto f_1 **no** es $\Omega(f_3)$. Un caso similar ocurre si comparanos f_3 con f_1.

Comparación de f_2 **con** f_3: Como f_2 es $\Theta(f_1)$, es decir que f_2 y f_1 son equivalentes asiontóticamente y f_1 no es ni $O(f_3)$ ni $\Omega(f_3)$ entonces tampoco f_2 es ni $O(f_3)$ ni $\Omega(f_3)$. De igual forma ocurre al comparar f_3 con f_2.

Comparación de f_3 **con** f_4: f_3 es $O(f_4)$ ya que:
$|f_3(n)| \leq k|f_4(n)| \iff |n| \leq k|n^3|$ se cumple para $k = 1$ y $n \geq 100$ cuando n es par y $|n^3| \leq k|n^3|$ se cumple para $k = 1$ y $n \geq 101$ cuando n es impar

f_3 **no** es $\Omega(f_4)$ ya que para un número par cualquiera mayor de 100 no se cumple que $|f_3| \geq k|f_4|$.

Comparación de f_4 **con** f_3: f_4 **no** es $O(f_4)$ ya que para un número par cualquiera mayor de 100 no se cumple que $|f_4| \leq k|f_3|$.

f_4 es $\Omega(f_3)$ ya que:
$|f_4(n)| \geq k|f_3(n)| \iff |n^3| \geq k|n|$ se cumple para $k = 1$ y $n \geq 100$ cuando n es par y $|n^3| \geq k|n^3|$ se cumple para $k = 1$ y $n \geq 101$ cuando n es impar

Análisis de f_5: $f_5(n) = ln(n^{n(2n)}) = ln(2n)ln(n) = (ln(2) + ln(n))ln(n) = ln(2)ln(n) + (ln(n))^2$ Esta función es $O(ln(n)^2)$ ya que:

$|f_5(n)| = |ln(2)ln(n) + (ln(n))^2| \leq |ln(n)^2 + ln(n)^2 = 2|ln(n)^2| \Rightarrow |f_5| \leq 2|ln(n)^2| \iff$

$$f_5(n) = O(ln(n)^2),$$

con $k = 2$ y para todo $n \geq 1$.
Esta función es asintóticamente menor que todas las demás funciones ya que una función logarítmica es siempre menor que una función lineal o polinómica en general. Esto lo podemos verificar de la siguiente forma, aplicando dos veces la regla de L'Hopital para comparar la función $ln(n)^2$ con n:

$$\lim_{n\to\infty} \frac{ln(n)^2}{n} = \lim_{n\to\infty} \frac{ln(n)^{2\prime}}{n'} = \lim_{n\to\infty} \frac{\frac{2ln(n)}{n}}{1} = \lim_{n\to\infty} \frac{2ln(n)}{n} = \lim_{n\to\infty} \frac{2ln(n)'}{n'} = \lim_{n\to\infty} \frac{\frac{2}{n}}{1} = \lim_{n\to\infty} \frac{2}{n} = 0$$

Por lo tanto n crece más rápido que $ln(n)^2$. También n^2 y n^3 acotarán asintóticamente a $ln(n)^2$

2. Sea $f : \mathbb{N} \mapsto \mathbb{R}$ tal que f es $O(n^{\frac{1}{2}})$. Si definimos la funcin $g : \mathbb{N} \mapsto \mathbb{R}$ tal que:

$$g(n) \begin{cases} f(n) + f(\frac{n}{2}) + ... + f(\frac{n}{2^i}) & \text{si } n = 2^i \text{ para } i \text{ en } \mathbb{N} \\ 0 & \text{en otro caso} \end{cases}$$

Concluya que g es $O(\sqrt{n}ln(n))$. Observación: la definición de g indica que esta función alcanza valores distintos de cero en potencias de 2.

3. Suponga $f : \mathbb{N} \mapsto \mathbb{R}^+, g : \mathbb{N} \mapsto \mathbb{R}^+, h : \mathbb{N} \mapsto \mathbb{R}^+, w : \mathbb{N} \mapsto \mathbb{R}^+$ tal que f es $O(g)$, h es $O(w)$ deduzca que:

   a. $f + h$ es $O(g + w)$
   b. $f * h$ es $O(g * w)$

Solución: Por hipótesis tenemos que si k y k' son constantes positivas entonces:

$$|f(x)| \leq k|g(x)|(i)$$

$$|h(x)| \leq k'|w(x)|(ii)$$

a. Si sumamos las hipótesis $(i), (ii)$: $|f(x)| + |h(x)| \leq k|g(x)| + k'|w(x)|$, como las funciones son siempre positivas entonces:

$$|f(x) + h(x)| \leq k|g(x)| + k'|w(x)| \leq k''|g(x)| + k''|w(x)|,$$

Con $k'' = max\{k, k'\}$. Si agrupamos las k'':

$$|f(x) + h(x)| \leq k''|g(x) + w(x)|,$$

Por lo tanto $f + h$ es $O(g + w)$, con $k'' = max\{k, k'\}$

b. Si multiplicamos las hipótesis $(i), (ii)$: $|f(x)||h(x)| \leq k|g(x)|k'|w(x)|$, por propiedad conmutativa y del valor absoluto nos queda:

$$|f(x)h(x)| \leq (kk')|g(x)w(x)|$$

$$|f(x)h(x)| \leq k''|g(x)w(x)|$$

Por lo tanto $f * h$ es $O(g * w)$, con $k'' = k * k'$

4. Suponga que $f : \mathbb{N} \mapsto \mathbb{R}$ es $O(n^{-\frac{1}{3}})$. Determine el $\lim_{x\to\infty} f(n)$

Solución: Por definición tenemos que $|f(n)| \leq k|g(n)|$, con k una constante positiva y $n \geq n_o$. Para este caso particular se traduce en: $|f(n)| \leq k|n^{-\frac{1}{3}}|$. Aplicándole el límite en ambos lados de la desigualdad tenemos que:

$$\begin{array}{ll}
\lim_{x\to\infty} |f(n)| \leq \lim_{x\to\infty} k|n^{-\frac{1}{3}}| & \\
\lim_{x\to\infty} |f(n)| \leq k * \lim_{x\to\infty} |\frac{1}{n^3}| & \text{linealidad del límite} \\
\lim_{x\to\infty} |f(n)| \leq k * 0 & \text{definición del límite} \\
\lim_{x\to\infty} |f(n)| \leq 0 &
\end{array}$$

$\Rightarrow \lim_{x\to\infty} f(n) = 0$

5. Muestre que el número de comparaciones $C(n)$ que realiza el procedimiento de ordenamiento por intercambio descrito a continuación es $O(n^2)$. Los elementos a ordenar son $a[1], a[2], ..., a[n]$..

Algoritmo 1 (Algoritmo de Ordenamiento)

1. *Hacer desde $i = 1$ hasta $i = n - 1$*
   2. *Hacer desde $j = i + 1$ hasta $j = n$*
      3. *Si $a[i] \leq a[j]$ entonces intercambiar $a[i]$ con $a[j]$,*

Solución: Analizando el algoritmo nos damos cuenta que el número de comparaciones que hace el algoritmo sobre una lista de n elementos, la cual denotaremos como $C(n)$, satisface la siguiente relación de recurrencia:

$$C(n) = C(n-1) + C(n-2) + ... + C(1)$$

El algoritmo ordena una lista de n elementos fijando un elemento y comparandolo con los elementos de listas más pequeñas de $n-1, n-2, ..., 1$ elementos. De esta forma $C(1) = 1$, ya para comparar el elemento fijo con una lista de longitud 1 se necesita sólo una comparación. También se cumple que $C(2) = 2, ..., C(n-2) = n-2, C(n-1) = n-1$. Por lo tanto, la cantidad de comparaciones necesarias para ordenar una lista de n elementos es la suma de los primeros $n-1$ enteros positivos.
La forma cerrada de la suma de los primeros n enteros positivos es:

$$1 + 2 + ... + n = \frac{n(n+1)}{2}$$

El número de comparaciones del algoritmo en cuestión llega hasta $n-1$, así que restamos n de ambos lados de la ecuación:

$$1 + 2 + ... + n - n = \frac{n(n+1)}{2} - n \iff 1 + 2 + ... + n - 1 = \frac{n^2 - n}{2}$$

Por lo tanto,

$$C(n) = \frac{n^2 - n}{2}$$

Verifiquemos que esta función es $O(n^2)$:

$$\begin{aligned} |C(n)| &= |\tfrac{n^2-n}{2}| \\ &= |2^{-1}(n^2 - n)| \\ &\leq |2^{-1}|(|n^2| + |n^2|) \\ &\leq 2^{-1}2|n^2| \\ &\leq |n^2| \end{aligned}$$

$\Rightarrow |C(n)| \leq |n^2| \iff C(n) = O(n^2)$, con $k = 1$ y $x \geq 0$