RELACIÓN DE EFICIENCIA

Hecho por Miguel Ángel Roldán Carmona

Problema 1.1. Probar que las siguientes sentencias son verdad:

- 17 es O(1)

En esta sentencia el valor de n esta elevado a la potencia de 0, por lo que realmente está sentencia podría expresarse alternativamente como $17*(n^0)$. Como sabemos, cualquier número elevado a 0 es igual a 1. Por tanto, está acotada asintóticamente por 1, siendo ese su orden.

- $\frac{n*(n-1)}{2}$ Es $O(n^2)$ y $\Omega(n^2)$ (es decir, $\Theta(n^2)$)

Esto quiere decir que tanto el tiempo teórico como el tiempo amortizado son iguales; es decir, tienden a n^2. Para hallar el orden de la sentencia primero descomponemos la fracción:

$\frac{n^2}{2}$ y $\frac{n}{2}$, la segunda de ellas en valor absoluto. Para probar cual de ellas es mayor basta con aplicar la regla de l'Hôpital comparando la primera con la segunda para ver que el resultado obtenido es ∞. Esto quiere decir que n^2 es el orden más grande, siendo por tanto el orden de la sentencia.

- $\max(n^3, 10\,n^2)$ es $O(n^3)$

Para probar esta sentencia simplemente basta con aplicar la regla de l'Hôpital comparando n^3 con $10n^2$. Se obtiene ∞ lo que quiere decir que el numerador (n^3) es mayor, siendo el orden igual a $O(n^3)$.

- $\text{Log}_2 n$ es $\Theta(\log_3 n)$

Para comprobar esta expresión vamos a utilizar la regla de l'Hôpital. Primero debemos pasar ambos logaritmos a una misma base. Por las propiedades de los logaritmos sabemos que $\log_2 n$ es igual a log(n)/log(2) y que $\log_3 n$ es igual a log(n)/log(3). Por tanto la fracción a aplicar la regla de l'Hôpital es la siguiente:

$$\frac{\log(n)*\log(3)}{\log(n)*\log(2)}$$

Como se puede observar los log(n) desaparecen y se quedan únicamente los logaritmos naturales en la expresión. Sabemos que aplicar la regla de l'Hôpital a una expresión de números naturales es igual a la expresión natural. log(3)/log(2) es ≈ 1,58. Por tanto, no podemos afirmar que la sentencia es cierta.

Problema 1.2. Encontrar el entero k más pequeño tal que f(n) es O(n^k) en los siguientes casos:

1. $f(n) = 13n^2 + 4n - 73$

Igualamos las expresiones $13n^2 + 4n - 73 = k*n^2 \rightarrow 13n^2 - kn^2 + 4n - 73 = 0 \rightarrow n^2(13 - k) + 4n = 73 \rightarrow n(n(13 - k)) + 4 = 73$. Despejamos la n que está dentro del paréntesis y obtenemos n= 69/(13-k).

Como sabemos que f(n) ϵ $O(n^2)$, k = 2, y n= 69/11.

2. f(n) = 1/(n +1)Igualamos las expresiones $1/(n + 1) = 1/k \rightarrow n + 1 = k \rightarrow n = k - 1$. Como sabemos que f(n) ϵ O(n), k = 1 y n = 0.

3. $f(n) = 1/(n - 1)$

Igualamos las expresiones $1/(n - 1) = 1/k \rightarrow n - 1 = k \rightarrow n = k + 1$. Como sabemos que $f(n) \in O(n)$, $k = 1$ y $n = 2$.

4. $f(n) = (n-1)^3$

$(n-1)^3 = n^3- 3n^2 + 3n - 1$. $n^3- 3n^2 + 3n - 1 = k*n^3 \rightarrow n (n (n (1 - k) - 3) + 3) - 1 = 0$

Despejamos la n que está dentro del paréntesis y obtenemos $n = 1 - 9/1-k =$

$1 + 9/k-1$. Como sabemos que $f(n) \in O(n^3)$, $k = 3$ y $n = 1 + 9/2 = 11/2$

5. $f(n) = (n^3 + 2n - 1)/(n + 1)$

6. $f(n) = (n^2 - 1)^{½}$

Problema 1.3. Ordenar de menor a mayor los siguientes órdenes de eficiencia:

$20000 < n\log2\log2(n^2) < n\log2(n^2) < \sqrt{n} < n < n + 100 < n^2 < n^3 + 1 < 3^{(\log2(n))} < 2^{(n)} < 3^{(n)} < 2^{(n)} + 3^{(n-1)} < n2^{(n)}$

Problema 1.4. Supongamos que $T1(n) \in O(f(n))$ y $T2(n) \in O(f(n))$. Razonar la verdad o falsedad de las siguientes afirmaciones:

a.- $T1(n) + T2(n) \in O(f(n))$.

Siendo T1(n) y T2(n) dos códigos independientes, se cumple la propiedad de la suma por la cual se cumple que $T1(n) + T2(n) \in$ a la O-grande mayor entre ellos. Como la O-grande es la misma, la afirmación es cierta.

b.- $T1(n) \in O(f(n^2))$

Sabemos que $T1(n) \in O(f(n))$, el cual es un subconjunto dentro de $f(n^2)$, por lo que la afirmación es cierta.

c.-$T1(n)/T2(n) \in O(1)$

Dos códigos, ya sean dependientes o independientes pueden producir una situación en la cual su O-grande se vea dividida entre ambos, por lo que esta afirmación es falsa-

Problema 1.5. Considerar las siguientes funciones de n:

$f1(n) = n^2$

$f2(n) = n^2 + 1000n$

$f3(n) = n$ si n es impar

n^3 si n es par

$f4(n) = n$ si $n \leq 100$

n^3 si $n > 100$

Indicar para cada par distinto i y j si f i(n) es O(f j(n)) y si f i(n) es Ω(f j(n))

Antes de empezar debemos saber cuáles van a ser las O-grandes de nuestras funciones:

f1(n) ϵ $O(n^2)$ y a $\Omega(n^2)$, f2(n) ϵ $O(n^2)$ y a $\Omega(n)$, f3 no podemos establecer una O-grande para esta función, ya que varía en el infinito, y esta función no la podemos comparar con el resto; f4(n) ϵ $O(n^3)$ pues tiende a esta función en el infinito y a $\Omega(n)$ mientras el valor de n no supere 100(sería el conjunto de los mejores casos) .

Procedemos a comparar:

f1(n) es O(f2(n)) pero no Ω(f2(n)), f1(n) es O(f4(n)) pero no Ω(f4(n)), f2(n) es O(f1(n)) y Ω(f1(n)), f2 es O(f4(n)) y Ω(f4(n)), f4(n) no es O(f1(n)) y si Ω(f1(n)), f4(n) no es O(f2(n)) y si Ω(f2(n)).

Problema 1.6. Demostrar que si f(n) ϵ O(g(n)) y g(n) ϵ O(h(n)) entonces f(n) ϵ O(h(n)).

Sabemos que g(n) es un subconjunto de h(n), por lo tanto como f(n) ϵ g(n) tambíen debe pertenecer a h(n) (Propiedad transitiva).

Problema 1.7. Demostrar que O(f(n)) ϵ O(g(n)) si f(n) ϵ O(g(n)) y g(n) ϵ O(f(n)).

Como puede verse, son dos funciones con la misma O-grande, pues están dentro del conjunto de O-grande de la otra función. Por tanto se cumple que O(f(n)) ϵ O(g(n)) y además se cumple que O(g(n)) ϵ O(f(n)) (Propiedad reflexiva).

Problema 1.8. Demostrar la siguiente jerarquía de órdenes de complejidad.

Para demostrar esta jerarquía basta con aplicar la regla de l'Hôpital sucesivamente hasta el final de la misma. Para ir comprobando cogeremos dos elementos consecutivos y comprobaremos que el resultado al aplicarle la regla de l'Hôpital es 0. Basta con que una de estas comparaciones sea mayor que 0 para que la jerarquía no se cumpla. En esta jerarquía se cumple que siempre sale 0 aplicando la regla de l'Hôpital hacia la derecha. Esta jerarquía es correcta.

Problema 1.9. Obtener usando la notación O-mayúscula la eficiencia del siguiente trozo de código

La eficiencia de este código es $O(n^3)$

Problema 1.10. Obtener usando la notación O-mayúscula la eficiencia de la siguiente función:

La eficiencia de esta función es $O(n^2)$

Problema 1.11. Obtener usando la notación O-mayúscula la eficiencia del siguiente trozo de código:

La eficiencia de este código es $O(n^2)$

Problema 1.12. Suponer que el parámetro n en la siguiente función es una potencia positiva de 2, es decir, n = 2, 4, 8, 16... Dar la fórmula que expresa el valor de la variable cont en términos del valor de n cuando la función termina.

La fórmula que expresa el valor de la variable cont cuando la función termina es $\log_2(n)$.

Problema 1.13. Estimar el peor caso de tiempo de ejecución para la siguiente función. Emplear la notación O-mayúscula.

La eficiencia de este código es O(2^n)

Problema 1.14. Dada la siguiente función:

1. ¿Cuál es el valor que devuelve la función?

Al tener varios return no se puede determinar el valor que retorna la función con exactitud.

2. Obtener una expresión para el peor caso de tiempo de ejecución de la función.

La eficiencia es O(log(n)).

Problema 1.15. Resolver la recurrencia siguiente en función de k y s:

$T(n) = k * T(n-1) + s^2 \qquad n > 1\ k,s > 1 \quad T(1)=1$

Problema 1.16. Considerando que el máximo de un vector es el máximo de dos valores:

1. El máximo de la primera mitad.
2. El máximo de la segunda mitad.

Implementar una función recursiva para calcular el máximo de un vector y estudiar su eficiencia.

Problema 1.17. Resolver la recurrencia.

Problema 1.18. Implementar una función recursiva que resuelva el problema de las Torres de Hanoi y estudiar la eficiencia en función de la altura de la torre.

Problema 1.19. Suponiendo que la función rectangle tiene un tiempo de ejecución constante, calcular la eficiencia de la siguiente función fractal en función de “n”, es decir, del número de niveles de profundidad.