"Análisis de Algoritmos"
José Helo Guzmán

Propiedades de la Notación Asintótica

Contenido

1. Introducción.

En el capítulo anterior hablamos de la BigOh, Omega y Theta. En este capítulo veremos algunas de sus propiedades e iniciaremos el proceso para encontrar el tiempo de ejecución de un algoritmo.

2. Propiedades de la Notación Asintótica.

Como todo conjunto de operaciones, la notación asintótica tiene ciertas cualidades que facilita su operación. Veamos algunas de estas propiedades.

3. Propiedad de los Polinomios.

Si $f(n) = a_0 + a_1*n + a_2*n^2 + ... + a_k*n^k$

$O(f(n)) = O(n^k)$

●● Ejemplo.

Si $f(n) = 7 + 8*n + 9* n^2 + 4*n^3$

$O(f(n)) = O(n^3)$

4. Propiedad Multiplicación de Constantes.

- Si f(n) = O(g(n)) entonces, a*f(n) = O(g(n)).

●● Ejemplo.

Si $f(n) = 2*n^2 + 5$

por lo tanto $f(n) = O(n^2)$

entonces

7*f(n)

$= 7*(2*n^2 + 5)$

$= 14*n^2 + 35$

$= O(n^2)$

5. Propiedad de la Adición.

Si a(n) = O(f(n))

y b(n) = O(g(n))

entonces

a(n) + b(n) = O(max(f(n) + g(n)))

●● Ejemplo.

Si

$a(n) = n + 5, \quad a(n) = O(n)$

$b(n) = n^2 + 7, \quad b(n) = O(n^2)$

Entonces:

$a(n) + b(n)$

$= (n + 5) + (n^2 + 7)$

$= n + 5 + n^2 + 7$

$= n^2 + n + 12$

$= O(\max(n,n^2))$

$= O(n^2)$

6. Propiedad de la Multiplicación.

Si

$a(n) = O(f(n))$

$b(n) = O(g(n))$

entoncex

$a(n) * b(n) = O(f(n)*g(n))$

Ejemplo.

Si

$a(n) = n + 5, \quad a(n) = O(n)$

$b(n) = n^2 + 7, \quad b(n) = O(n^2)$

Entonces:

$a(n) * b(n)$

$= (n + 5) * (n^2 + 7)$

$= n^3 + 7*n + 5*n^2 + 35$

$= n^3 + 5*n^2 + 7*n + 35$

$= O(n * n^2)$

$= O(n^3)$

7. Propiedad de la Construcción de Theta.

Si $f(n) = O(g(n))$

y $f(n) = \Omega(g(n))$

entonces

$f(n) = \theta(g(n))$

●● Ejemplo.

$f(n) = n^2 + 5$

$f(n) = O(n^2)$

$f(n) = \Omega(n^2)$

entonces

$f(n) = \theta(n^2)$

8. Cálculo del Tiempo de Ejecución de un Programa.

A continuación enunciamos algunas de las reglas básicas cuando se utiliza la notación de la gran O.

●● Definición de O(1)

Si $f(n) = 4$

entonces

$f(n) = O(4) = O(1)$ denota una instrucción simple.

●● Manejo de las constantes

Si $f(n) = 2*n$

entonces

$O(2*n) = O(n)$

en general:

$O(c*n) = O(n)$

●● Suma de términos:

Si $f(n) = 1 + 2*n + 1$

entonces

$O(1 + 2*n + 1) = O(2*n + 2) = O(n)$

●● Más sumas de términos.

Si

$f(n) = O(100*n)$

$g(n) = O(n^2)$

Entonces:

$f(n) + g(n)$

$= O(100*n) + O(n^2)$

$= O(100*n + n^2)$

$= O(n^2 + 100*n)$

$= O(n^2)$

Otra forma de indicarlo es:

$f(n) + g(n)$

$= O(\max(n , n^2))$

$= O(n^2)$

●● Multiplicación de términos.

Si

$f(n) = O(100*n)$

$g(n) = O(n^2)$

Entonces:

$f(n) * g(n)$

$= O(100*n) * O(n^2)$

$= O(100*n * n^2)$

$= O(100*n^3)$

$= O(n^3)$

9. Comparación de Funciones.

De manera matemática, dadas dos funciones, se puede conocer cuál crece más que la otra de la siguiente manera.

Sea $f(n) = f1(n) + f2(n)$

Si

$$\text{límite} \left(\frac{f1(n)}{f2(n)} \right) = 0$$

Entonces $f(n)$ es $O(f2(n))$

Sin embargo, este método es muy tedioso. Por ello utilizaremos alguno de los tres métodos que se presentan a continuación.

●● Ejemplo.

Tenemos:

$f(n) = \log(n)$

$g(n) = n$

Formamos el límite:

$$\underset{n \to +\infty}{\text{límite}} \left(\frac{\log(n)}{n} \right)$$

Usamos L’Hopital:

$$\underset{n \to +\infty}{\text{límite}} \left(\frac{(1/n)}{1} \right)$$

$$\underset{n \to +\infty}{\text{límite}} \left(\frac{1}{n} \right) = 0$$

Por lo tanto:

$n > \log(n)$

▬▬ 10. Utilizando una Demostración.

El uso de logaritmos puede ayudar para comparar dos funciones.

●● Recordemos algunas fórmulas de logaritmos.

Cada una de las reglas se pone un indicador al principio para futuras referencias.

(f1) $\log(a*b) = \log(a) + \log(b)$

(f2) $\log(a/b) = \log(a) - \log(b)$

(f3) $\log(a^b) = b*\log(a)$

(f4) $a\hat{}(\log_k b) = b\hat{}(\log_k a)$

(f4) $a^{(\log_k b)} = b^{(\log_k a)}$

(f5) Si $a^b = n$, entonces $b = \log_a(n)$

●● Ejemplo.

Sean

$f(n) = n^2$

$g(n) = n^3$

Al aplicar logaritmo en ambas funciones:

n^2	n^3
$\log(n^2)$	$\log(n^3)$
$2*\log(n)$	$3*\log(n)$

Tenemos que : $2*\log(n) < 3*\log(n)$

Si $n > 1$: $2 < 3$

Por lo tanto: $n^2 < n^3$

●● Ejemplo de Funciones.

Sea

$f(n) = n^2 * \log(n)$

$g(n) = n * (\log(n))^{10}$

Veamos cuál es mayor.

$f(n)=$	$g(n)=$
$n^2 * \log n$	$n * (\log n)^{10}$
$\log(n^2 * \log n)$	$\log(n * (\log n)^{10})$
$\log n^2 + \log(\log n)$	$\log n + \log(\log n)^{10}$
$2*\log n + \log(\log n)$	$\log n + 10*\log(\log n)$

Y podemos ver que:

$2*\log n + \log(\log n) > \log n + 10*\log(\log n)$

Esto es, pues $\log(\log n)$ es muy pequeño.

$2*\log n > \log n$

11. Utilizando una Tabla.

Supongamos que deseamos conocer cuál de estas funciones es mayor:

$f(n) = n^2$

$g(n) = n^3$

Obviamente sabemos cuál de las dos es mayor, pero supongamos que no lo sabemos.

●● Ejemplo.

Tenemos:

$f(n) = n^2$

$g(n) = n^3$

n	n^2	n^3
2	4	8
3	9	27
4	16	25
5	25	125
6	36	216
7	49	343

Se puede observar $f(n) < g(n)$

12. Utilizando un Gráfico.

Una forma muy eficiente y elegante de determinar cuál es la mayor de dos funciones es por medio de un gráfico.

Sea

f(n) = 3*n^(√n)

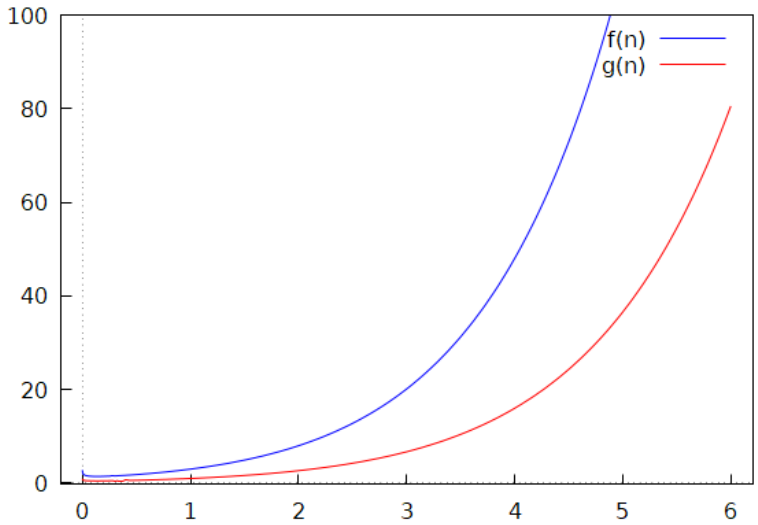
g(n) = 2^(√n * log n)

Veamos cuál es mayor.

» Gráfico.
» Comparación de f(n) y g(n)

```
load(draw)$

log2(x):=log(x)/log(2);
f(n):= 3*n^(sqrt(n));
g(n):= 2^(sqrt(n) * log2(n));


ini:0;
fin:6;
wxdraw2d
(
   xrange = [ini-0.20,fin+0.20],
   yrange = [ini-0.20,100],
   grid=false,
   xaxis=true,
   yaxis=true,

   color=blue,
   key="f(n)",
   explicit( f(n) ,n,ini,fin),

   color=red,
   key="g(n)",
```

```
    explicit(g(n), n,ini,fin)

)$
```



13. El Mejor Caso, Peor Caso y el Promedio.

En los algoritmo que se presentan a continuación, se hablará siempre de estos tres escenarios.

El peor caso, significa que se considera la situación en que el algoritmo debe utilizar la mayor cantidad de operaciones.

El mejor caso, significa que se considera la situación en que el algoritmo debe utilizar la menor cantidad de operaciones.

El caso promedio significa que se consideran todos los posibles casos, se asume que son igualmente probables (suposición que no siempre es correcta) y se obtiene su promedio. Esta forma de estimar un algoritmo suele ser difícil de obtener. Además en muchas ocasiones coincide con el peor caso posible.

14. Resumen.

- Se han estudiado las propiedades de la notación asintótica.

- Se han mostrado ejemplos de cómo utilizar la notación asintótica para obtener resultados.

- Se han comparado diferentes funciones para establecer cuál es mayor. En particular el método de graficar las funciones es muy sencillo y práctico.

- Se ha hablado del pero caso, el mejor caso y el caso promedio.

15. Ejercicios.

Ejercicio 1.

Indique cuál de las siguientes funciones es mayor.

$f(n) = n^{(\log n)}$

$g(n) = 2^{(\sqrt{n})}$

Ejercicio 2.

Indique cuál de las siguientes funciones es mayor.

$f(n) = 2^{(\log n)}$

$g(n) = n^{(\sqrt{n})}$

Ejercicio 3.

Indique cuál de las siguientes funciones es mayor.

$f(n) = 2^n$

$g(n) = 2^{(2*n)}$

Ejercicio 4.

Indique si cada uno de los enunciados que se presentan son verdaderos o falsos

- $(n + 5)^m = O(n^m)$
- $2^{(n+1)} = O(2^n)$
- $2^{(2*n)} = O(2^n)$
- $\sqrt{(\log n)} = O(\log \log n)$
- $n^{(\log n)} = O(2^n)$