Asesoría 1: Análisis Asintótico

May 2021

1 Propiedades/Definiciones Útiles

- $log_a(a) = 1$
- $log_a(1) = 0$
- $log_a(a^x) = x$
- $log(a * b) = log(a) + log(b)$
- $log(\frac{a}{b}) = log(a) - log(b)$
- $log_x(a^b) = b * log_x(a)$
- $log(\sqrt[b]{a}) = \frac{log(a)}{b}$
- $x^{log_b(a)} = a^{log_b(x)}$
- $a^{log_a(x)} = x$
- $log_b(\frac{1}{a}) = -log_b(a)$
- $log_b(a) = \frac{1}{log_a(b)}$
- $a^{-n} = \frac{1}{a^n}$
- $(a^m)(a^m) = a^{m+n}$
- $(a^m)^n = a^{mn}$
- $(\frac{a}{b})^m = \frac{a^m}{b^m}$
- $(ab)^m = a^m b^m$
- $\frac{a^m}{a^n} = a^{m-n}$
- $x - 1 < \lfloor x \rfloor \leq x$
- $x \leq \lceil x \rceil < x + 1$
- $\frac{1}{\infty} = 0$

- $\frac{0}{0}$ = indeterminado
- $\frac{\infty}{\infty}$ = indeterminado
- **L'Hôpital**: $\lim_{x \to c} \frac{f(x)}{g(x)} = \lim_{x \to c} \frac{f'(x)}{g'(x)}$. Si en primera instancia el límite da indeterminado, entonces aplicar L'Hôpital derivando el numerador y el denominador. Luego, evaluar nuevamente el límite. Sin embargo, algunos límites pueden dar indeterminado a primera vista, pero si se les aplica algo como una ley de exponentes por ejemplo, ya no se vuelven indeterminados. Por ejemplo, $\lim_{x \to \infty} \frac{n^2}{n^3}$. Si reemplazamos n por ∞, obtenemos $\frac{\infty}{\infty}$ lo cual es indeterminado y nos podríamos sentir tentados a usar L'Hôpital, pero no es necesario, ya que aplicando la ley de exponentes, obtenemos: $\lim_{x \to \infty} \frac{1}{n} = 0$. No tuvimos que aplicar L'Hôpital.
  Cabe resaltar que L'Hôpital no se tiene que aplicar una sola vez, se puede aplicar más de una vez. Por ejemplo, si $\frac{f'(x)}{g'(x)}$ sigue dando indeterminado, entonces podemos intentar con $\frac{f''(x)}{g''(x)}$ y así sucesivamente.

2 Límites

1. $\lim_{x \to \infty} \frac{f(n)}{g(n)} \neq 0, \infty$ entonces $f(n) = \Theta(g(n))$
   - *$f(n)$ es aproximadamente igual a $g(n)$.*
2. $\lim_{x \to \infty} \frac{f(n)}{g(n)} \neq \infty$ entonces $f(n) = O(g(n))$
   - *$f(n)$ es aproximadamente menor o igual a $g(n)$.*
3. $\lim_{x \to \infty} \frac{f(n)}{g(n)} \neq 0$ entonces $f(n) = \Omega(g(n))$
   - *$f(n)$ es aproximadamente mayor o igual a $g(n)$.*
4. $\lim_{x \to \infty} \frac{f(n)}{g(n)} = 1$ entonces $f(n) \sim g(n)$
   - *$f(n)$ es casi igual a $g(n)$.*
5. $\lim_{x \to \infty} \frac{f(n)}{g(n)} = 0$ entonces $f(n) = o(g(n))$
   - *$f(n)$ es aproximadamente menor a $g(n)$.*
6. $\lim_{x \to \infty} \frac{f(n)}{g(n)} = \infty$ entonces $f(n) = \omega(g(n))$
   - *$f(n)$ es aproximadamente mayor a $g(n)$.*
7. Notar que hay solapamiento entre ciertas definiciones. Por mencionar una, si cumplimos el caso 5, también cumplimos el caso 2, lo cual tiene sentido, ya que si $f(n) = o(g(n))$, también $f(n) = O(g(n))$.

3 Términos

3.1 Asintóticamente Mayor/Menor

Podemos decir que $f(n)$ es **asintóticamente mayor** que $g(n)$ sí se cumple la siguiente condición:

$$\frac{f(n)}{g(n)} = \Omega(n^{\epsilon})$$
$$\frac{f(n)}{g(n)} = cn^{\epsilon}$$

Para $c > 0$ y $\epsilon > 0$. Esto se lee de la siguiente manera: *el cociente entre $f(n)$ y $g(n)$ es un factor de n^{ϵ}* o también que *$f(n)$ es asintóticamente mayor que $g(n)$ por un factor de n^{ϵ}*.

El caso para **asíntóticamente menor** es análogo. En el caso anterior, se podría decir que *$g(n)$ es asintóticamente menor que $f(n)$*.

3.2 Polinómicamente Mayor

Podemos decir que $f(n)$ es **polinómicamente mayor** a $n^{log_b a}$ si:

$$f(n) = \Omega(n^{log_b a + \epsilon})$$

o si:

$$\lim_{x \to \infty} \frac{f(n)}{n^{log_b a + \epsilon}} \neq 0$$

3.3 Polinómicamente Menor

Podemos decir que $f(n)$ es **polinómicamente menor** a $n^{log_b a}$ si:

$$f(n) = O(n^{log_b a - \epsilon})$$

o si:

$$\lim_{x \to \infty} \frac{f(n)}{n^{log_b a - \epsilon}} \neq \infty$$

4 Árbol de Recursión

Como mencionado en clase, hay tres casos en el árbol de recursión. A continuación se mostrará de manera genérica la complejidad del algoritmo a *grosso modo*. Cabe resaltar que la idea detras de cada forma genérica es la siguiente: **por cada nivel del árbol de recursión, sumo el costo total de cada nivel**.

1. El costo total de cada nivel se mantiene **igual**. Un ejemplo clásico es el árbol de recursión de *Merge Sort*. En este caso, el costo total de cada nivel siempre es cn.

   - Sea c_i el costo total de el i-ésimo nivel del árbol, es claro que $c_0 = c_1 = c_2 = ... = c_h$, entonces podríamos ponerle un solo nombre, digamos *costo*. En este caso, podríamos usar la siguiente sumatoria genérica para obtener una idea para descubrir la complejidad del algoritmo:

$$\sum_{i=0}^{h} costo$$

$$= h * costo$$

2. El costo total de cada nivel **decrece**. Un ejemplo sería, $T(n) = 3T(\frac{n}{4}) + \Theta(n^2)$. En este caso, el costo total de cada nivel va decreciendo mientras nos acercamos a las hojas. El primer paso sería identificar el ratio r por el cual el costo n^2 decrece en cada nivel ($r < 1$, porque decrece). Esto se puede hacer dibujando el árbol de recursión. (Ejercicio: dibujar el árbol de recursión de la recurrencia descrita al comienzo del párrafo e identificar ese ratio r).

   - Sea r el ratio y $f(n)$ el costo por combinar y dividir (esa es la definición formal, en el ejemplo anterior, $f(n) = n^2$). Entonces la sumatoria sería:

$$\sum_{i=0}^{h} r^i * f(n)$$

$$= f(n) * \sum_{i=0}^{h} r^i$$

$$= f(n) * \frac{1 - r^h}{1 - r}$$

     También podríamos considerar que:

$$\sum_{i=0}^{h} r^i * f(n)$$

$$\leq \sum_{i=0}^{\infty} r^i * f(n)$$

$$= f(n) * \sum_{i=0}^{\infty} r^i$$

$$= f(n) * \frac{1}{1 - r}$$

Nótese que el término $-r^h$ del numerador se hace 0, esto se debe a que si $h = \infty$ y $r < 1$, un decimal multiplicado infinitas veces por sí mismo, tiende a ser 0, por lo que $-r^h$ se haría 0 y solo quedaría el 1 solito en el numerador.

3. El costo total de cada nivel **crece**. En ejemplo sería, $T(n) = 3T(\frac{n}{2}) + \Theta(1)$. En este caso, el costo total de cada nivel va aumentando mientras nos acercamos a las hojas. Al igual que el caso 2, el primer paso sería identificar el ratio x por el cual el costo $f(n)$ aumenta ($x > 1$, porque aumenta).
   - Sea x el ratio y $f(n)$ el costo por combinar y dividir. Entonces la sumatoria sería:
$$\sum_{i=0}^{h} x^i * f(n)$$
$$= f(n) * \frac{x^{h+1} - 1}{x - 1}$$

5 Método de Substitución

- $\lfloor \frac{n}{2} \rfloor$
  - Considerarlo como par, es decir, $\frac{n}{2}$ si se quiere mostrar un O.
  - Considerarlo como impar, es decir, $\frac{n}{2} - 1$ si se quiere mostrar un Ω.
- $\lceil \frac{n}{2} \rceil$
  - Considerarlo como par, es decir, $\frac{n}{2}$ si se quiere mostrar un Ω.
  - Considerarlo como impar, es decir, $\frac{n}{2} + 1$ si se quiere mostrar un O.
- **Probando** O:
  - En el último paso, se debe tener la siguiente estructura para asegurarse que la inducción se ha cerrado:
$$\leq assumption - term$$
    donde
$$term \geq 0$$
- **Probando** Ω:
  - En el último paso, se debe tener la siguiente estructura para asegurarse que la inducción se ha cerrado:
$$\geq assumption + term$$
    donde
$$term \geq 0$$
- Para el caso de o y ω, es lo mismo, pero con $<$ y $>$, respectivamente.

6 Teorema Maestro

Digamos que queremos saber si $T(n) = 4T(\frac{n}{2}) + n^2lg(n)$ se puede resolver mediante el teorema maestro. Tenemos que descubir cómo se compara $n^2lg(n)$ y $n^{log_b(a)} = n^2$. Para ello, sacaremos el límite de $f(n)$ entre n^2:

$$\lim_{n\to\infty} \frac{n^2lg(n)}{n^2}$$

$$= \lim_{n\to\infty} \log(n)$$

$$= \infty$$

Lo cual implica que:

$$f(n) = n^2lg(n) = \omega(n^2)$$

Por lo tanto, nos inclinaremos por el tercer caso del teorema maestro. Ahora que ya sabemos que $f(n)$ es asintóticamente mayor a $n^{log_b(a)}$, tenemos que probar que también es polinómicamente mayor a $n^{log_b(a)}$, para ello evaluaremos el siguiente límite:

$$\lim_{n\to\infty} \frac{n^2lg(n)}{n^{log_b(a)+\epsilon}}$$

$$= \lim_{n\to\infty} \frac{n^2lg(n)}{n^{2+\epsilon}}$$

$$= \lim_{n\to\infty} \frac{lg(n)}{n^{\epsilon}}$$

Lo cual da indeterminado. Aplicando L'Hôpital:

$$= \lim_{n\to\infty} \frac{\frac{1}{n}}{\epsilon n^{\epsilon-1}}$$

$$= \lim_{n\to\infty} \frac{n^{-1}}{\epsilon n^{\epsilon-1}}$$

$$= \lim_{n\to\infty} \frac{1}{\epsilon n^{\epsilon}}$$

$$= 0$$

Como el límite es igual a 0. Eso implica que:

$$f(n) = n^2lg(n) = o(n^{2+\epsilon})$$

Pero nosotros esperabamos:

$$f(n) = n^2 lg(n) = \Omega(n^{2+\epsilon})$$

Por lo que no podemos aplicar el teorema maestro. Pero nótese que tratando de comprobar lo de polinómicamente mayor, obtuvimos un upper bound $o(n^{2+\epsilon})$ para $T(n)$. Ya que $\epsilon > 0$, digamos que $\epsilon = 1$, entonces podríamos decir que $T(n) = o(n^3)$.