Taller 1 - Algoritmos

Jonathan Campo Rangel.

1) [Cormen09]

a)(3.1-2) Mostrar que para cualquier reales a y b, con $b>0$; $(n+a)^b = \Theta(n^b)$.

Si $|a| \leq n \Rightarrow (n+a) \leq 2n$

así $n \geq 2a$

Si $|a| \leq \frac{n}{2} \Rightarrow (n+a) \geq \frac{n}{2}$

Entonces:

$$0 \leq \frac{n}{2} \leq (n+a) \leq 2n \Rightarrow 0 \leq \frac{n^b}{2^b} \leq (n+a)^b \leq 2^b(n^b)$$

Utilizando constantes c_1, c_2

$$0 \leq c_1 n^b \leq (n+a)^b \leq c_2 n^b \Rightarrow c_1 = \left(\frac{1}{2}\right)^b, \ c_2 = 2^b, \ n_0 = 2a$$

Entonces $(n+a)^b = \Theta(n^b)$.

b)(3.1-7) Probar que $o(g(n)) \cap \omega(g(n))$ es $\{\}$ (vacío)

Por contradicción:

$o(g(n)) \cap \omega(g(n)) \neq \{\} \Rightarrow \forall\ c_1, c_2 > 0$ se tiene:

$$0 \leq c_1 g(n) < f(n) < c_2 g(n)$$

donde $n \geq \max(n_1, n_2)$

Con $c_1 = c_2$ llegamos a una contradicción, se prueba $o(g(n)) \cap \omega(g(n)) = \{\}$

c)(3.3)

a) Ordenar funciones por taza de crecimiento.

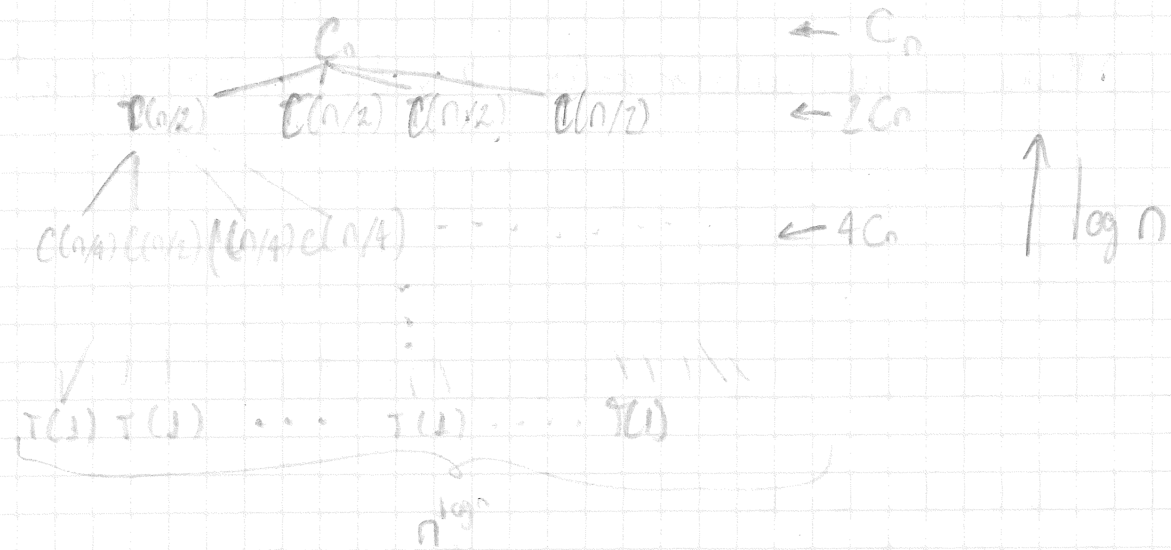
- $2^{2^{n+1}}$ $\quad 2^{2^n}$
- $(n+1)!$ $\quad n!$
- e^n $\quad n \cdot 2^n$ $\quad 2^n$ $\quad (3/2)^n$
- $(\lg n)^{\lg n} = n^{\lg \lg n}$ $\quad (\lg n)!$
- n^3 $\quad n^2 = 4^{\lg n}$ $\quad n \lg n = \lg(n!)$ $\quad 2^{\lg n} = n$ $\quad (\sqrt{2})^{\lg n} = \sqrt{n}$ $\quad 2^{\sqrt{2 \lg n}}$
- $\lg^2 n$ $\quad \lg n$ $\quad \sqrt{\lg n}$ $\quad 2^{\lg^* n}$
- $\lg \lg n$
- $\lg^* n$ $\quad \lg^*(\lg n)$ $\quad \lg(\lg^* n)$
- $n^{1/\lg n} = 2$ $\quad 1$
- $2^{\sin n}$

$^*(\lg n)^{\lg n} = n^{\lg \lg n} \Leftrightarrow a^{\log_b c} = c^{\log_b a}$

$n^{1/\lg n} = 2 \Leftrightarrow n^{1/\lg n} = 2^{\lg n \cdot 1/\lg n} = 2$

b) Función tal que $f(n) \neq O(g_i(n)) \neq \Omega(g_i(n))$, $= 2^{2\cos n}$

d) (4.4-7) Dibujar arbol de recurrencia para $T(n) = 4T(\lfloor n/2 \rfloor) + cn$



e) Metodo Maestro para acotar

- $T(n) = 8T(n/2) + n$
- $T(n) = 8T(n/2) + n^3$
- $T(n) = 8T(n/2) + n^5$

- $T(n) = 8T(n/2) + n$;

$$\log_2 8 = 3, \quad n^3 > n \Rightarrow T(n) = \Theta(n^3)$$

- $T(n) = 8T(n/2) + n^3$; $\log_2(8) = 3$, $n^3 = n^3 \Rightarrow T(n) = \Theta(n^3 \log n)$

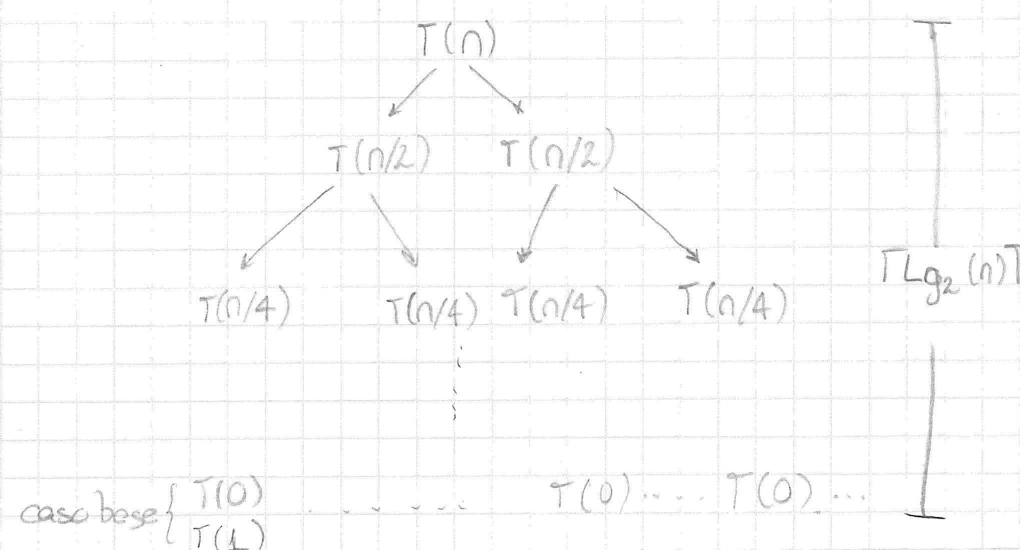
- $T(n) = 8T(n/2) + n^5$;

$$\log_2 8 = 3, \quad n^3 < n^5 \Rightarrow T(n) = \Theta(n^5)$$

2) a) Ecuación de recurrencia para $T(n)$ de misterio(n)

$$
\begin{cases} T(n) = 2T(n/2) & \text{, para } n \neq 0 \neq 1 \\ 1 & \text{, para } n = 0,\ n = 1 \end{cases}
$$

b) Arbol de Recursión



I) Altura: $\lceil \lg_2(n) \rceil$

II) Nodos por nivel: 2^n

III) Suma de nodos por nivel: $T(n)$

IV) Suma total: $\lceil \lg_2(n) \rceil \cdot T(n)$

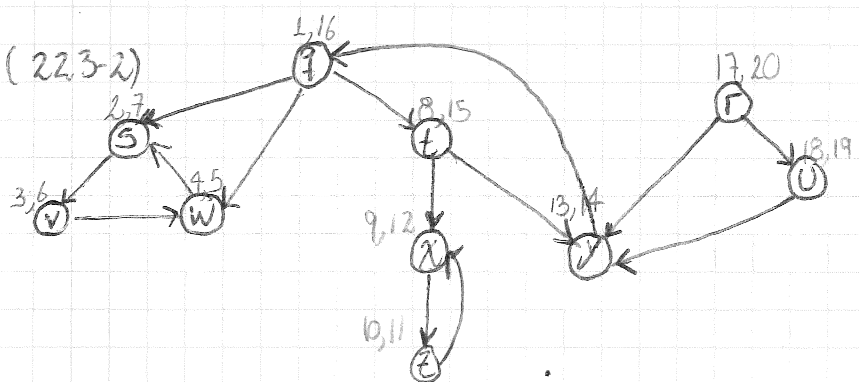
3) (22.3-1) Tabla 3x3. Grafo dirigido, búsqueda en profundidad

Dirigido (i, j)	Blanco	Gris	Negro
Blanco	TBFC	BC	C
Gris	TF	TFB	TFC
Negro	—	B	TFBC

Grafo no dirigido, Búsqueda en Profundidad

No dirigido (i, j)	Blanco	Gris	Negro
Blanco	TB	TB	—
Gris	TB	TB	TB
Negro	—	TB	TB

4) (22.3-2)



Búsqueda de profundidad (orden alfabético)

Aristas de arbol: (q, s), (s, v), (v, w), (q, t), (t, x), (x, z), (t, y), (r, u)

Aristas Traseras: (w, s), (z, x), (y, q)

Aristas Delanteras: (q, w)

Aristas Cruzadas (r, y), (u, y)

5(22.4-2): Algoritmo de encontrar caminos (contar)

Pseudocode

```
Find_Paths(G, t, s) {
  vertex = [vertices de G]
  t vertex = 1
  vertex[t] = 1
  vertex[i] = 0   // Diferente de t

  if (DFS(t, i))
    vertex[i]++

  return: vertex[s]
}
```