$1 < \log n < n < n \log n < n^2 < n^3 < \underbrace{2^n < n!}_{\text{intratables.}}$ DAA Repaso.

REPASO SERIES.

1. Aritmética

$$\sum_{i=1}^{n} i = \boxed{S(n) = \frac{(n+1)\,n}{2}}$$

"El primero más el último por $n/2$"

2. Binomial al cuadrado.

$$\sum_{i=1}^{n} i^2 = \boxed{S(n) = \frac{(2n+1)\,n\,(n+1)}{6}}$$

3. Geométrica.

$$S_i = \sum_{i=0}^{n-1} r^i = \boxed{S(n) = \frac{r^{0} - r^n}{1-r}}$$

4. Mezcla Aritmética y Geométrica.

$$\sum_{i=1}^{n} i \cdot r^i = \boxed{\frac{n r^{n+2} - (n+1) r^{n+1} + r}{(r-1)^2}}$$

"Donde r puede ser un número cualquiera"

5. Aproximación por integrales - Cotas.

5.1. Monótona creciente.

$$\int_{m-1}^{n} f(x)\,dx \leq \sum_{i=m}^{n} f(i) \leq \int_{m}^{n+1} f(x)\,dx$$

5.2. Monótona decreciente.

$$\int_{m}^{n+1} f(x)\,dx \leq \sum_{i=m}^{n} f(i) \leq \int_{m-1}^{n} f(x)\,dx$$

NOTACIONES ASINTÓTICAS

1. COTA SUPERIOR O $\leq c \cdot n^x$

$$f(n) \in O(g(n)) = \{f(n) : \exists c > 0 \text{ y } n_0 > 0 \;/\; 0 \leq f(n) \leq c \cdot g(n),\ \forall n \geq n_0\}$$

2. COTA INFERIOR Ω $\geq c \cdot n^x$

$$\Omega(g(n)) = \{f(n) : \exists c > 0 \text{ y } n_0 > 0 \;/\; 0 \leq c \cdot g(n) \leq f(n),\ \forall n \geq n_0\}$$

3. COTA AJUSTADA Θ (Usar ambas)

$$\Theta(g(n)) = \{f(n) : \exists c_1 > 0,\ c_2 > 0 \text{ y } n_0 > 0 \;/\; 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)\ \forall n > n_0\}$$

4. Con límites.

- $f(x) \in O(g(n)) \Leftrightarrow \lim_{n \to \infty} \frac{f(n)}{g(n)} < \infty$ ser constante o cero.
- $f(n) \in \Omega(g(n)) \Leftrightarrow \lim_{n \to \infty} \frac{f(n)}{g(n)} > 0$ ser constante o infinito
- $f(n) \in \Theta(g(n)) \Leftrightarrow \lim_{n \to \infty} \frac{f(n)}{g(n)} =$ constante > 0

INSERCIÓN EN BUCLES.

$$T_{BUCLE} = 1 + \sum (1 + T + 1) + 1$$

Diferenciar mejor y peor caso.

EFICIENCIA DE ALGORITMOS RECURSIVOS.

$$T(n) \begin{cases} \text{caso base} \\ \text{caso recursivo.} \end{cases}$$

1. Coger el recursivo.
2. Expandir hasta llegar a general.
3. El caso base llega al general.
4. Sustituir en el general.
5. Sacar complejidad.

- No simplificar las expresiones, poner como potencias.
- En caso de sumatorio, va hasta n si es $=$ sino $n-1$

MÉTODO GENERAL DE RESOLUCIÓN DE RELACIONES DE RECURRENCIA.

1. Recurrencias Homogéneas: Las que pueden igualarse a 0 (Todo Términos $T(n)$)

1.1. Sacar Polinomio característico.

1.2. Factorizar, Sacar sus raíces

1.2.1. Si todas las raíces son distintas (Ninguna Repetida)

$$T(n) = C_1 \cdot r_1^n + C_2 \cdot r_2^n + \ldots + C_k \cdot r_k^n$$

r_k: raíces

C_k: Constantes. Se sacan con los casos base.

1.2.2. Si son raíces múltiples (Repetidas)

$$T(x) = (x - r_1)^{m_1} (x - r_2)^{m_2} \ldots (x - r_s)^{m_s}$$

m_s: multiplicidad de la raíz.

$$T(n) = C_1 \cdot P_1(n) \cdot r_1^n + C_2 \cdot P_2(n) \cdot r_2^n + \ldots + C_s \cdot P_s(n) \cdot r_s^n$$

2. Recurrencias no Homogéneas: Las no igualables a 0. Términos con no T

$$T(n) + a_1 T(n-1) + \ldots + a_k T(n-k) = \sum_{i=1}^{s} P_i^{d_i}(n)\, b_i^{\,n}$$

↳ Polinomio por potencia de grado d_i

2.1. Sacar Polinomio característico de los términos $T(n)$

2.2. Cada $P_i^{d_i}(n)\, b_i^{\,n}$ genera un $(x - b_i)^{d_i+1}$

2.3. Factorizar ambos lados.

2.4. Resolver como casos anteriores, dependiendo si se repite o no.

(*) Para hallar las constantes sacar tantos casos base como constantes y sustituir. (Máximo sistema de 3 variable)

TEOREMA MAESTRO. (En algoritmos de DIVIDE y VENCERÁS)

$$T(n) \begin{cases} c & n=1 \\ aT(n/b) + dn^k & n \geq 1 \end{cases}$$

a: Nº Subprogramas

b: Tamaño de los Subprogramas.

Entonces

$$T(n) \in \begin{cases} \text{Si } \frac{a}{b^k} < 1 & \text{entonces } O(n^k) \\ \text{Si } \frac{a}{b^k} = 1 & \text{entonces } O(n^k \log n) \\ \text{Si } a/b^k > 1 & \text{entonces } O(n^{\log_b a}) \end{cases}$$

—— o ——

$$T(n) \begin{cases} \text{Complejidad Caso Base} \\ a \cdot T(n-b) + \text{Complejidad Recursiva} \end{cases}$$

⊛ Debe ser n/b

a: Nº llamadas recursivas

b: Deterioro del problema.

—— o —— SUMATORIOS DESDOBLADOS

$$\sum_{i=m}^{n} i = \sum_{i=1}^{n} i - \sum_{i=1}^{m-1} i$$

—— o ——

En expansión de recurrencias los sumatorios de geométricas hasta $i-1$

Fraccionarios → Cambio Variable DAA.

Examen Parcial 2015-2016

$$\begin{cases} T(1) = 1 \\ T(n) = T(n/2) + \log_2 n \end{cases}$$

1.) Cambio de variable y cambio de función $n/b \Rightarrow n = b^k$

$\boxed{n = 2^k}$

$T(n) = T(2^k) = t(k)$ $T(2^k) = t(k)$

$T(2^k) = T(2^k/2) + \log_2 2^k$ (Parcial de esto)

$T(2^k) = T(2^{k-1}) + k$ ← Solo se cambia $T(\_)$ lo de dentro. Notación $t(k)$

2.) Cambio de función.

$t(k) = t(k-1) + k$

$t(k) - t(k-1) = k \cdot 1^k$

3.) Polinomio y raices.

$(x-1)\,(x-1)^2 \longrightarrow (x-1)^3$

4.) Resolver. Raices Multiples.

$t(n) = C_1 \cdot 1^k + C_2 \cdot k \cdot 1^k + C_3 \cdot k^2 \cdot 1^k$

5.) Hallar los constantes (Casos base 3) → $n = 2^k$ $T(2^k) = t(k)$

$t(0) = T(2^0) = 1 \longrightarrow t(0)$

$t(1) = T(2^1) = 2 \longrightarrow t(1)$

$t(2) = T(2^2) = 4 \longrightarrow t(2)$

6.) Hallar constantes.

$t(0) = C_1 = 1$

$t(1) = C_1 + C_2 + C_3 = 2$

$t(2) = C_1 + 2C_2 + 2C_3 = 4$

$C_1 = 1$

$C_2 = 1/2$

$C_3 = 1/2$

7.) Solución $k = \log_2 n$

$t(k) = 1 + 1/2\, k + 1/2\, k$

$\boxed{T(n) = 1 + 1/2 \log_2 n + 1/2 \log_2 n \in O(\log n)}$