ANÀLISIS D'ALGORITMES

Complexitat d'Algoritme: Recursos computacionals que consumeix (Texec i Espai Mem)

$\Theta(n) \to$ Límit Asimtòtic Exacte ≡ "tan ràpid com..."

$O(n) \to$ Límit Asimtòtic Inferior ≡ "no més ràpid que..."

$\Omega(n) \to$ Límit Asimtòtic Superior ≡ "no tant ràpid com..."

$\underbrace{\log(n)}_{\text{Logarítmic}} \ll \sqrt{n} \ll \underbrace{n}_{\text{Lineal}} \ll \underbrace{n \cdot \log(n)}_{\text{Casi-lineal}} \ll \underbrace{n^2}_{\text{Quadràtic}} \ll \underbrace{n^3}_{\text{Cúbic}} \ll \underbrace{K^n}_{\text{Exponencial}} \ll n! \ll n^n$ || $\Theta(1) \leftarrow$ Constant

Propietats (f, g funcions)

- Operacions elementals cost $\Theta(1)$. Suma, ... ; Assignacions ; Esc/Lec ; Comparació ; Accés.
- $\boxed{\Theta(f) + \Theta(g) = \Theta(\max\{f, g\})}$ "Mana l'algoritme més costós"
- $\boxed{\Theta(f) \cdot \Theta(g) = \Theta(f \cdot g)}$ # Igual per $O()$, $\Omega()$ # Comú en bucles
- $c > 0 \to O(f) = O(c \cdot f)$
- Passar param per valor o return de vector $\to \Theta(n)$ n é quant de param o v.size()
- Passar param per referència $\to \Theta(1)$ Pq estem dient només on està la variable
- Podem fer canvi de base: $\boxed{\log_c(x) = \dfrac{\log_b(x)}{\log_b(c)}}$

$$\boxed{\lim_{n \to \infty} \frac{f(n)}{g(n)} \begin{cases} 0 \to f < g \\ c \to f = g \\ \infty \to f > g \end{cases}}$$

Algoritmes Recursius

temps de la recursivitat

$$\boxed{T(n) = a \cdot T(n-c) + f(n)}$$

nº crides recursives — Part no recursiva

$$\boxed{T(n) = a \cdot T(n/b) + f(n)}$$

Teorema (1)

$$T(n) = \begin{cases} g(n) & \text{si } 0 \leq n < n_0 \\ a \cdot T(n-c) + f(n) & \text{si } n \geq n_0 \end{cases}$$

Cost del Cas Base

Condició del cas base. n_0 és una constant que marca si s'entra o no.

Master theorem (1)

Si $T(n)$ és de la forma $T(n) = \begin{cases} g(n) & \text{if } 0 \leq n < n_0 \\ a \cdot T(n-c) + f(n) & \text{if } n \geq n_0 \end{cases}$

nº crides recursives en el mateix nivell

Cost part no recursiva

I $f(n) = \Theta(n^k)$ per $k \geq 0$. Llavors tenim:

$$T(n) = \begin{cases} \Theta(n^k) & \text{si } a < 1 \\ \Theta(n^{k+1}) & \text{si } a = 1 \\ \Theta(a^{n/c}) & \text{si } a > 1 \end{cases}$$

Això és molt important.

Master theorem (2)

Si $T(n)$ és de la forma $T(n) = \begin{cases} g(n) & \text{if } 0 \leq n < n_0 \\ a \cdot T(n/b) + f(n) & \text{if } n \geq n_0 \end{cases}$

I $f(n) = \Theta(n^k)$ per $k \geq 0$. Definim $\alpha = \log_b a$. Tenim:

$$T(n) = \begin{cases} \Theta(n^k) & \text{si } \alpha < k \;||\; a < b^k \\ \Theta(n^k \log n) & \text{si } \alpha = k \;||\; a = b^k \\ \Theta(n^\alpha) & \text{si } \alpha > k \;||\; a > b^k \end{cases}$$