Tema 1: Eficiencia

Órdenes de eficiencia →
- Constante $O(1)$
- Lineal $O(n)$
- Cuadrático $O(n^2)$
- Logarítmico $O(\log(n))$
- Exponencial $O(a^n)$
- ...

Análisis de algoritmos (Teórico)

- Sentencias simples $x += 1 \Rightarrow O(1)$
- Sentencias condicionales

  if (Ev Condicion) {
  Bloque 1
  } else {
  Bloque 2
  }

  $\Rightarrow$ máx{ O(Ev. Condición), O(B1), O(B2) }

- Sentencias repetitivas

  Mientras (Condición), hacer: { h(n) veces
  Bloque → f(n)

  $g(n)$

  $\Rightarrow O(g(n) + h(n)(g(n) + f(n)))$

- Secuencia de sentencias

  Sentencia 1
  Sentencia 2
  Sentencia S

  $\Rightarrow$ máx{ $O(S_1), O(S_2) \ldots O(S_s)$ }

Eficiencia de una función recursiva

Se calcula el tiempo de ejec $T(n)$ como ecuación en recurrencias

Ejemplo: factorial(n) {
if (n ≤ 1) return 1;
else return n*factorial(n-1)

$$\Rightarrow T(n) = \begin{cases} 1 & n \le 1 \\ 1 + T(n-1) & n > 1 \end{cases}$$

Resolución:
- → Desarrollo en serie "a lo bruto"
- → Ecuación característica
  - → Ec. Lineales $T(n) = a_1 \cdot T(n-1) + a_2 T(n-2) + \ldots + a_k \cdot T(n-k)$

    $T(n) = t_n = x^n$

    * No homogéneas $\Rightarrow p(x) = b^n q(x)$ $\quad gr(q) = d$ (Como si fuera homogéneo)

      $\Rightarrow (x-b)^{d+1}$

  - → No lineales $\Rightarrow$ Cambio de variable $\Rightarrow n = 2^m, 3^m \ldots \log(m)$

    $m = \log_2(n), \log_3(n) \ldots e^n$

Tipos de órdenes de eficiencia

- Peor caso, umbral superior $\Rightarrow O(\cdot)$
- Mejor caso, umbral inferior $\Rightarrow \Omega(\cdot)$
- Orden exacto $\Rightarrow$ Si $O(f(n)) = \Omega(f(n)) \Rightarrow \Theta(f(n)) = O(f(n))$ ($= \Omega(f(n))$)