

# Análisis Recursivo

Análisis y Diseño de Algoritmos

Dr. Jaime Osorio Ubaldo

# Ecuación de recurrencia

El tiempo de ejecución de un algoritmo recursivo se expresa por medio de una ecuación de recurrencia.

**Ejemplo** Para calcular el factorial de un número. El algoritmo recursivo es

```
función factorial(n)  
  si (n = 0)  
    retorna 1  
  sino  
    retorna n * factorial(n - 1)
```

Si  $c_1$  es el costo temporal del caso base, la ecuación de recurrencia,  $f(n)$ , es

$$f(n) = \begin{cases} c_1, & n = 0 \\ f(n-1) + c_2, & n > 0 \end{cases}$$

**Ejercicio** Determine la complejidad de este algoritmo.

# Solución

Sabemos que

$$f(n) = f(n-1) + c_2$$

de aquí

$$f(n-1) = f(n-2) + c_2 \text{ y}$$

$$f(n-2) = f(n-3) + c_2$$

con lo cual

$$f(n) = f(n-3) + 3c_2$$

podemos deducir que

$$f(n) = f(n-i) + i \cdot c_2$$

para el caso base

$$n - i = 1$$

luego

$$f(n) = c_1 + (n-1)c_2$$

Por lo tanto el algoritmo de complejidad  $\mathcal{O}(n)$ .

Dada la siguiente ecuación de recurrencia de un determinado algoritmo .

$$T(n) = \begin{cases} 1, & n \leq 1 \\ T(\frac{n}{2}) + 1, & n > 1 \end{cases}$$

Determine la complejidad de este algoritmo.

Dada la siguiente ecuación de recurrencia de un determinado algoritmo .

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

Determine la complejidad de este algoritmo.