

Programación Avanzada

OMAR S. GÓMEZ

Octubre, 2016



Diseño de algoritmos recursivos

El primer paso consiste en identificar un algoritmo recursivo para resolver el problema en cuestión.

Descomponer el problema de tal forma que su solución quede definida en función de ella misma pero para un tamaño menor (caso general o recursivo) así como la tarea a realizar para un caso simple (caso base o reducido).

Los casos generales siempre deben avanzar hacia un caso base. Es decir, la llamada recursiva se convierte en un subproblema más pequeño (hasta llegar al caso base).

Así pues para resolver algún problema a través de un algoritmo recursivo se deben diseñar: casos base, casos generales y la solución en términos de ellos.

Ejemplo

Ejemplo de definición recursiva para el cálculo del factorial de un entero positivo:

$$n! = \begin{cases} 1 & \text{si } n = 0 \\ n \cdot (n - 1)! & \text{si } n > 0 \end{cases}$$

Diagram illustrating the recursive definition of factorial with annotations:

- Condición que induce al caso base**: Points to the condition $si\ n = 0$.
- Condición que ayuda a acotar el problema**: Points to the condition $si\ n > 0$.
- Caso general**: Points to the recursive formula $n \cdot (n - 1)!$.

Algunas pautas para el diseño de algoritmos recursivos

¿Hay una salida no recursiva del algoritmo (caso base)?, ¿El algoritmo cumple con la especificación en este caso?

¿Cada llamada recursiva se refiere a un caso más pequeño del problema original?

Suponiendo que la(s) llamada(s) recursivas cumplen la especificación (son correctas), ¿cumple la especificación el algoritmo completo?

Actividad I (individual)

Diseña un algoritmo recursivo que calcule la potencia de un número natural entero positivo x^n

Implementa el algoritmo en el lenguaje de programación de tu preferencia

Ejecuta también el algoritmo en SREC (implementado en Java)

Actividad II (individual)

Diseña un algoritmo recursivo que calcule la suma de dos números enteros positivos

Implementa el algoritmo en el lenguaje de programación de tu preferencia

Ejecuta también el algoritmo en SREC (implementado en Java)

Actividad III (individual)

Diseña un algoritmo recursivo que calcule el producto de dos números enteros positivos

Implementa el algoritmo en el lenguaje de programación de tu preferencia

Ejecuta también el algoritmo en SREC (implementado en Java)