Recursividad

Un procedimiento o función recursiva es aquella que se llama a sí misma. esta característica permite a un procedimiento recursivo repetirse con valores diferentes de parámetros. la recursión es una alternativa a la iteración muy elegante en la resolución de problemas, especialmente si estos tienen naturaleza recursiva.

Normalmente, una solución recursiva es menos eficiente en términos de tiempo de computadora que una solución iterativa debido al tiempo adicional de llamada a procedimientos.

En muchos casos, la recursión permite especificar una solución más simple y natural para resolver un problema que otro caso sería difícil. Por esta razón la recursión (recursividad) se una herramienta muy potente para la solución de problemas y la programación.

Ejemplos de caso de recursividad, secuencia de números Fibonacci

Esta serie numérica es un ejemplo típico de cumplimiento de las dos propiedades generales de todo procedimiento recursivo. La serie de fibonacci es:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34…

El término inicial es $a_0 = 0$, y los siguientes son $a_1 = 1$, $a_2 = 2$, $a_3 = 3$, $a_4 = 5$, $a_5 = 8$… observándose que cada término es la suma de los dos términos precedentes excepto $a_0 = 0$ y $a_1 = 1$. La serie puede ser definida recursivamente del modo siguiente:

Fibonacci (n) = n si n = 0 o n = 1
Fibonacci (n) = Fibonacci (n - 2) + Fibonacci (n - 1) para n > 1

Esta definición recursiva hace referencia a ella misma dos veces. Y la condición par que dejen de hacerse llamadas recursiva es que n sea 0 o 1. la llamada recursiva se hace en términos menores de n, n - 2, n - 1, por lo que se va acercando a los valores de qué depende la condición de terminación.

Referencia:

Luis Joyane Aguilar, Ignacio Zahonero Martínez. (1998). Estructura de datos. España: McGraw-Hill.