

**Instituto Tecnológico de Tijuana**

**Recursividad**

**Materia:**

Estructura de Datos

**Profesor(a):**

Ray Brunett Parra Galaviz

**Alumno(a):**

Jiménez Mayoral Gloria Alejandra – 17212146

**Fecha:**

06 de septiembre de 2018

**T6. 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 éstos 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 en otro caso sería difícil. Por esta razón la recursión (recursividad) es una herramienta muy potente para la resolución de problemas y la programación.

Un procedimiento o función recursivos han de cumplir dos propiedades generales para no dar lugar a un bucle infinito con las sucesivas llamadas:

* Cumplir una cierta condición o criterio base del que dependa la llamada recursiva.
* Cada vez que el procedimiento o función se llamen a sí mismos, directa o indirectamente, debe estar más cerca del incumplimiento de la condición de que depende la llamada.

Cuando **no** utilizar la recursividad

La solución recursiva de ciertos problemas simplifica mucho la estructura de los programas. Como contrapartida, en la mayoría de los lenguajes de programación las llamadas recursivas a procedimientos o funciones tienen un coste de tiempo mucho mayor que sus homólogos iterativos. Se puede, por tanto, afirmar que la ejecución de un programa recursivo va a ser más lenta y menos eficiente que el programa iterativo que soluciona el mismo problema, aunque, a veces, la sencillez de la estructura recursiva justifica el mayor tiempo de ejecución.

La recursión se puede presentar de dos maneras diferentes:

1. **Directa:** el programa o subprograma se llama directamente a sí mismo.
2. **Indirecta:** el subprograma llama a otro subprograma, y éste en algún momento, llama nuevamente al primero.

En toda definición recursiva de un problema siempre se deben establecer dos pasos diferentes y muy importantes; el paso básico y el paso recursivo. El primero, uno o varios, se utiliza como condición parada o fin de la recursividad. A éste llegamos cuando encontramos la solución del problema o cuando decidimos que ya no vamos a seguir, porque no está dadas las condiciones para hacerlo. El paso segundo, propicia la recursividad. Se pueden presentar uno o varios, nuevamente dependiendo del problema a resolver.

Cuando se analiza la solución recursiva de un problema es importante determinar con precisión cuales serán los pasos básico y recursivo. En cada vuelta del ciclo es importante que nos acerquemos cada vez más a la solución del problema, o sea, al paso básico.

**Bibliografía**

* Joyanes A, L., Zahonero M, I. *Estructura de datos. Algoritmos, abstracción y objetos*. España: McGraw Hill.
* Cairó, O., Guardati, S. (2006). Estructuras de Datos (3ª edición). México: McGraw-Hill.