

Recursividad

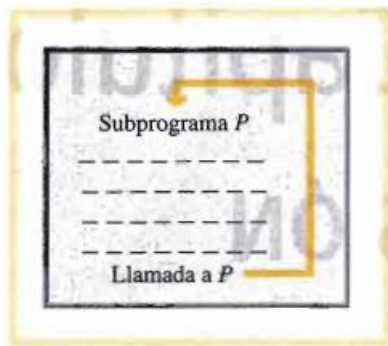
La recursión o recursividad es un recurso muy poderoso que permite expresar soluciones simples y naturales a ciertos tipos de problemas. Es importante considerar que no todos los problemas son naturalmente recursivos; algunos sí lo son y otros no.

Una función recursiva es aquella que se llama a sí mismo, bien directamente o bien indirectamente a través de otra función. La recursividad es un tópico importante examinado frecuentemente en cursos en los que se trata de resolución de algoritmos y en cursos relativos a Estructuras de Datos.

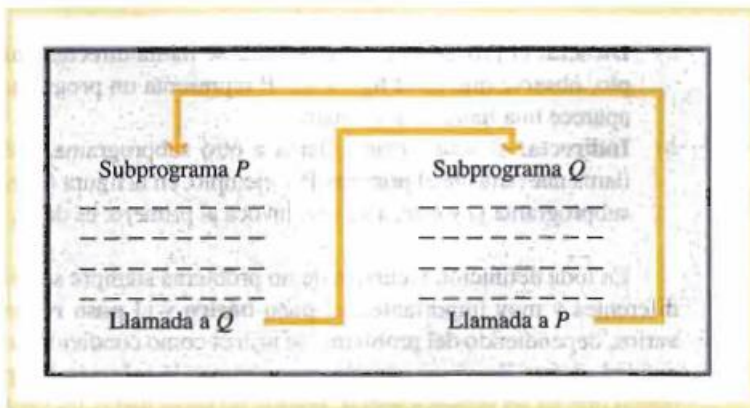
Un objetivo recursivo es aquel que aparece en la definición de sí mismo, así como el que se llama a sí mismo. Los árboles, por ejemplo, representan las estructuras de datos, no lineales y dinámica, más eficientes que existen actualmente en computación. Es decir, en cualquier actividad de programación que se realice con árboles se utiliza la recursividad.

La recursión se puede presentar en dos maneras diferentes:

-Directa: el programa o subprograma se llama directamente a sí mismo.



-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, dependiendo del problema, se utiliza como condición de parada o fin de la recursividad. A éste llegamos cuando encontramos la solución del problema o cuando decimos que ya no vamos a seguir, porque no están dadas las condiciones para hacerlo. El paso segundo, por otra parte, 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 cuáles serán los pasos básicos y recursivo.

Recursión infinita

La iteración y la recursión pueden producirse infinitamente. Un bucle infinito ocurre si la prueba o test de continuación de bucle nunca se vuelve false; una recursión infinita ocurre si la etapa de recursión no reduce el problema en cada ocasión de modo que converja sobre el caso base o condición de salida.

En realidad la recursión infinita significa que cada llamada recursiva produce otra llamada recursiva y ésta a su vez otra llamada recursiva y así para siempre. En la práctica dicha función se ejecutará hasta que la computadora agota la memoria disponible y se produce una terminación anormal del programa.

El flujo de control de una función recursiva requiere tres condiciones para una terminación normal:

- ° Un test para detener (o continuar) la recursión (condición de salida o caso base).
- ° Una llamada recursiva (para continuar la recursión).
- ° Un caso final para terminar la recursión.

Referencias:

-Cairó, O., Guardati, S., (2006). *Estructura de Datos*. McGraw Hill

- Joyanes,L., Sánchez, L., Zahonero, I., (2007). *Estructura de datos en C++*. McGraw Hill