Síntesis conceptual

Asignatura: Programación
Unidad: 11. Recursividad y complejidad algorítmica

Resumen

La recursividad es otra técnica de desarrollo que permite resolver ciertos problemas con un flujo ciclico que de forma iterativa visto hasta ahora sería dificil de realizar. La recursividad consiste en dividir el problema en partes mas pequeñas y con sentencias de código sencillas y faciles encapsuladas en un mismo método. La principal caracteristica de la recursividad es que en el mismo cuerpo del método **existe una o más llamada al propio método**, a consecuencia, es necesario definir una condición de terminación o **caso base** entre llamada y llamada al mismo método, si no, se produciria un bucle infinito de llamadas. Las ventajas de su uso es que hace más sencillo el algoritmo, solo se puede emplear en ciertos patrones y evita el uso excesivo de sentencias de control o bucles.

Existen 4 tipos de recursividad, **Simple** solo existe una sola llamada al propio método. **Múltiple** existe dos o más llamadas. **Cruzada o indirecta**, hay dos métodos con una llamada al otro y viceversa. **Anidada**, uno de los parametros de la llamada es la propia llamada.

La complejidad algoritmica estudia la cantidad de recursos que puede consumir una solución al algoritmo, la forma de medir es mediante la notación Big-0 (Notación Asintótica o Notación Landau). La notación Big-o establece unos valores por defecto que es necesario conocer cuando se aplica según el codigo del algoritmo, para ello es necesario tener unos conocimiento matematicos avanzados, aún así se establecen unas reglas sencillas que facilitan encontrar que orden de complejidad le corresponde a un código.

Conceptos fundamentales

- **Caso base:** Parte de la recursividad que en código se representa con una sentencia de control y haciendo que termine el flujo de llamadas.
- **Notación Big-O:** La unidad de medida para estudiar el consumo de recursos de la complejidad algoritmica. Se suele representar de la forma O( ).
- **Búsqueda dicotómica:** También conocida como búsqueda binaria, algoritmo que busca un elemento diviendo en dos partes y por cada parte las vuelve a dividir, su flujo se representa en arbol de dos ramificaciones.
- **Quick sort:** Algoritmo de ordenación rápida que elige un pivote o referencia y empieza a ordenar por cada lado del pivote. Es un tipico algoritmo para realizar con recursividad.
- **Merge sort:** Algoritmo de ordenación, divide la colección en dos partes iguales, con los elementos mezclados los va uniendo en una unica lista ordenada.