

Clase 7. Algoritmo - Diagrama de Flujo

Concepto de Algoritmo

Un **algoritmo** es una secuencia finita, ordenada y bien definida de instrucciones que resuelven un problema o realizan una tarea específica. En términos más formales, un algoritmo es un conjunto de pasos que, al ejecutarse, transforman un conjunto de entradas en un conjunto de salidas. Esta transformación está diseñada para ser reproducible, y debe llevarse a cabo de manera eficiente y sin ambigüedades.

Las tres partes de un algoritmo

Un algoritmo consta de tres elementos fundamentales:

1. **Entrada:** Los datos iniciales sobre los que opera el algoritmo.
2. **Proceso:** El conjunto de operaciones que transforman los datos de entrada.
3. **Salida:** Los resultados generados tras ejecutar el algoritmo.

Historia del concepto

El término "algoritmo" proviene de la latinización del nombre del matemático persa **Al-Juarismi**, quien en el siglo IX introdujo muchos de los métodos algebraicos que conocemos hoy. Los algoritmos han sido fundamentales en el desarrollo de las matemáticas, y su formalización moderna comenzó a tomar forma en el siglo XX con el trabajo de matemáticos como Alan Turing y Kurt Gödel, quienes investigaron la naturaleza de los procesos computacionales.

Características de un Algoritmo

Los algoritmos poseen una serie de características clave que garantizan su efectividad y utilidad:

- **Finitud:** Un algoritmo debe terminar después de un número finito de pasos.
- **Determinismo:** Cada paso del algoritmo debe estar claramente definido y no debe dar lugar a ambigüedades.
- **Eficiencia:** El algoritmo debe resolver el problema de manera eficiente, utilizando la menor cantidad posible de recursos computacionales.
- **Entrada y salida:** Debe haber al menos una entrada y una salida claramente definidas.
- **Generalidad:** Debe poder aplicarse a un conjunto de problemas de la misma naturaleza, no solo a un caso particular.

Tipos de Algoritmos

Algoritmos de Búsqueda

Son algoritmos diseñados para encontrar un elemento específico dentro de una estructura de datos.

- **Búsqueda Secuencial:** Recorre todos los elementos de una lista hasta encontrar el deseado. Es simple pero ineficiente para listas grandes.
- **Búsqueda Binaria:** Solo funciona en listas ordenadas. Divide la lista en dos y descarta la mitad en cada paso, logrando así una mayor eficiencia.

Algoritmos de Ordenamiento

Estos algoritmos reordenan los elementos de una lista en función de un criterio determinado, como el valor numérico o alfabético.

- **Ordenamiento de Burbuja:** Compara pares de elementos adyacentes y los intercambia si están en el orden incorrecto.
- **Ordenamiento por Selección:** Busca el elemento más pequeño en cada iteración y lo coloca en su posición final.

Algoritmos Voraces

Estos algoritmos toman decisiones locales óptimas con la esperanza de encontrar una solución global óptima.

- **Problema de la Mochila Fraccional:** Se seleccionan los objetos que maximizan el valor total sin superar un peso dado.
- **Algoritmo de Dijkstra:** Encuentra el camino más corto entre dos nodos en un grafo.

Programación Dinámica

Esta técnica resuelve problemas dividiéndolos en subproblemas más pequeños y almacenando sus resultados para evitar recálculos.

- **Serie de Fibonacci:** Un ejemplo clásico donde los valores previos se almacenan para evitar cálculos redundantes.

Algoritmos Probabilísticos

Se basan en la aleatoriedad para llegar a una solución.

- **Algoritmo de Montecarlo:** Utiliza simulaciones aleatorias para resolver problemas numéricos.
- **Algoritmo de Las Vegas:** Produce siempre una respuesta correcta, pero el tiempo de ejecución es aleatorio.

Otros tipos de Algoritmos

- **Numéricos:** Resuelven problemas que involucran cálculos numéricos.
- **No numéricos:** Operan con datos no numéricos, como cadenas de texto.
- **Gráficos:** Utilizan diagramas de flujo o grafos para representar la lógica del algoritmo.
- **No gráficos:** Se representan mediante pseudocódigo, una forma simplificada del lenguaje de programación.

Simbología en los algoritmos

Los algoritmos a menudo utilizan símbolos que representan diferentes tipos de operaciones:

- **Aritméticos:** (+, -, *, /) para operaciones matemáticas.
- **Relacionales:** (>, <, ==) para comparar valores.
- **Lógicos:** (&&, ||, !) para combinar condiciones lógicas.

Ejemplos de Algoritmos en la Vida Real

Los algoritmos no se limitan al ámbito de la informática; tienen aplicaciones prácticas en la vida cotidiana:

- **Recetas de cocina:** Son algoritmos que indican qué pasos seguir para preparar un platillo.
- **Manuales:** Proporcionan secuencias de instrucciones para ensamblar o usar productos.
- **Operaciones matemáticas:** Como el método de multiplicación o división.

Diagramas de Flujo

Un **diagrama de flujo** es una representación gráfica de un algoritmo o proceso. Usa una serie de símbolos conectados por flechas para describir el flujo de las operaciones de manera visual.

Tipos de Diagramas de Flujo

Existen diversos tipos de diagramas de flujo que se adaptan a diferentes necesidades:

- **Vertical:** Los procesos se representan de arriba hacia abajo.
- **Horizontal:** Los pasos se distribuyen de izquierda a derecha.
- **Panorámico:** Se utilizan varias direcciones, permitiendo una representación más compleja.
- **Arquitectónico:** Representa procesos relacionados con la planificación y diseño estructural.

Pasos para la Construcción de un Diagrama de Flujo

1. **Análisis:** Comprender el problema y los pasos necesarios para resolverlo.
2. **Construcción:** Traducir esos pasos a un formato visual utilizando los símbolos adecuados.
3. **Prueba de Escritorio:** Verificar el diagrama de flujo simulando su ejecución con valores de entrada y comprobando que los resultados sean los esperados.

Simbología en los Diagramas de Flujo

- **Círculo:** Indica un conector que une partes de un diagrama.
- **Flecha:** Muestra la dirección del flujo del proceso.
- **Óvalo o Elipse:** Representa el inicio o el fin del proceso.
- **Rectángulo:** Indica una operación o acción.

- **Rombo:** Muestra una decisión que bifurca el flujo en dos o más caminos.
- **Triángulo:** Usado en algunos casos para indicar almacenamiento de datos.
- **Triángulo Inverso:** Representa un proceso de extracción de datos o eliminación.

Diferencia entre Algoritmo y Diagrama de Flujo

Un algoritmo es una secuencia de instrucciones precisas para resolver un problema, mientras que un diagrama de flujo es una representación gráfica de ese algoritmo. Los diagramas de flujo son útiles para visualizar la estructura lógica de un proceso, pero no reemplazan la necesidad de escribir el algoritmo en un lenguaje de programación o pseudocódigo para su ejecución.