
Algoritmos

Algoritmos, Pseudocódigo y diagramas de flujo.



Curso de Programación en Java
Juan Francisco Maldonado León

| | |
|--|----|
| Algoritmos | 3 |
| <i>Pseudocódigo</i> | 6 |
| <i>Diagramas de flujos</i> | 7 |
| Concepto de Variable..... | 9 |
| <i>Nombres de Variables</i> | 9 |
| <i>Contenido y asignación de contenido a variables</i> | 11 |

Algoritmos

Es un conjunto prescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos que no generen dudas a quien deba realizar dicha actividad. Dados un estado inicial y una entrada, siguiendo los pasos sucesivos se llega a un estado final y se obtiene una solución.

La palabra algoritmo deriva del nombre de un matemático árabe del siglo IX, llamado Al-Khuwarizmi, quien estaba interesado en resolver ciertos problemas de aritmética y describió varios métodos para resolverlos. Estos métodos fueron presentados como una lista de instrucciones específicas (como una receta de cocina) y su nombre es utilizado para referirse a dichos métodos.

Se espera que un algoritmo tenga varias propiedades. La primera es que un algoritmo no debe ser ambiguo, o sea, que si se trabaja dentro de cierto marco o contexto, cada instrucción del algoritmo debe significar sólo una cosa.

Imaginemos que debemos especificar como determinar si el número 7317 es un número primo.

Algoritmo 1: Divida el 7317 entre sus anteriores buscando aquellos que lo dividan exactamente.

Algoritmo 2: Divida el número 7317 entre cada uno de los números 1, 2, 3, 4, ..., 7315, 7316. Si una de las divisiones es exacta, la respuesta es no. Si no es así, la respuesta es sí.

El algoritmo 1 no especifica claramente cuáles son los valores a lo que se refiere, por lo que resulta ambiguo.

El algoritmo 2 presenta una solución no ambigua para este problema. Existen otros algoritmos mucho más eficaces para dicho problema, pero esta es una de las soluciones correctas.

Resumiendo, un algoritmo puede definirse como una secuencia ordenada de pasos elementales, exenta de ambigüedades, que lleva a la solución de un problema dado en un tiempo finito.

Para comprender totalmente la definición anterior falta clarificar que se entiende por "**paso elemental**".

Escriba un algoritmo que permita preparar una tortilla de papas de tres huevos. El enunciado anterior basta para que un cocinero experto lo resuelva sin mayor nivel de detalle, pero si este no es el caso, se deben describir los pasos necesarios para realizar la preparación.

Esta descripción puede ser:

Mezclar papas cocidas, huevos y una pizca de sal en un recipiente

Freír

Esto podría resolver el problema, si el procesador o ejecutor del mismo no fuera una persona que da sus primeros pasos en tareas culinarias, ya que el nivel de detalle del algoritmo presupone muchas cosas. Si este problema debe resolverlo una persona que no sabe cocinar, se debe detallar, cada uno de los pasos mencionados, pues estos no son lo bastante simples para un principiante.

De esta forma, el primer paso puede descomponerse en:

Pelar las papas

Cortarlas en cuadraditos

Cocinar las papas

Batir los huevos en un recipiente

Agregar las papas al recipiente y echar una pizca de sal al mismo

El segundo paso (freír) puede descomponerse en los siguientes tres:

Calentar el aceite en la sartén

Verter el contenido del recipiente en la sartén

Dorar la tortilla de ambos lados

Nótese además que si la tortilla va a ser realizada por un niño, algunas tareas (por ejemplo batir los huevos) pueden necesitar una mejor especificación. El ejemplo anterior sólo pretende mostrar que la lista de pasos elementales que compongan nuestro algoritmo depende de quién sea el encargado de ejecutarlo. Si en particular, el problema va a ser resuelto utilizando una computadora, el conjunto de pasos elementales conocidos es muy reducido, lo que implica un alto grado de detalle para los algoritmos.

Se considera entonces como un **paso elemental** aquel que no puede volver a ser dividido en otros más simples. De ahora en adelante se utilizará la palabra instrucción como sinónimo de paso elemental.

Pseudocódigo

Es una descripción de alto nivel compacta e informal del principio operativo de un programa informático u otro algoritmo.

Utiliza las convenciones estructurales de un lenguaje de programación real, pero está diseñado para la lectura humana en lugar de la lectura mediante máquina, y con independencia de cualquier otro lenguaje de programación.

Diagramas de flujos

Es la representación gráfica de un algoritmo.

Simbología y significado

Óvalo o Elipse: Inicio y Final (Abre y cierra el diagrama).

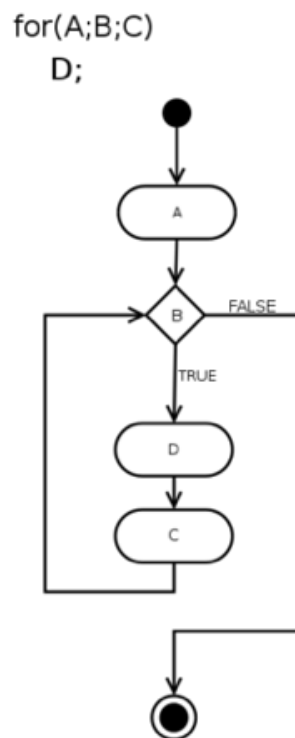
Rectángulo: Actividad (Representa la ejecución de una o más actividades o procedimientos).

Rombo: Decisión (Formula una pregunta o cuestión).

Círculo: Conector (Representa el enlace de actividades con otra dentro de un procedimiento).

Triángulo boca abajo: Archivo definitivo (Guarda un documento en forma permanente).

Triángulo boca arriba: Archivo temporal (Proporciona un tiempo para el almacenamiento del documento).



Concepto de Variable

Las variables son estructuras de datos que pueden cambiar de contenido a lo largo de la ejecución de un programa. Estas estructuras corresponden a un área reservada en la memoria principal de la computadora.

Nombres de Variables

Una variable es un espacio de la memoria del ordenador a la que asignamos un contenido que puede ser un valor numérico (sólo números, con su valor de cálculo) o alfanumérico (sólo texto o texto con números). Cada variable tiene un único nombre el cual no puede ser cambiado. Dos o más variables pueden tener el mismo contenido, pero no el mismo nombre.

El nombre de una variable comenzará siempre por una letra, pudiendo contener a continuación tanto letras como números.

Las letras pueden ser tanto mayúsculas como minúsculas. No se admiten nombres de variables incluyendo espacios en blanco ni símbolos especiales como guiones, puntos, comas, comillas, etc. ni símbolos matemáticos ni palabras clave (que veremos más adelante, y que incluyen "inicio", "fin", "verdadero", "falso", "entonces"...). El nombre de una variable será lo suficientemente largo como para impedir que pueda confundirse con otra variable por tener nombre similar, así como para aportar una indicación de cuál es el contenido o función que cumple.

Ejemplos: Nombre de variables.

| Nombre de variable | Comentarios |
|---|---|
| Numerodeplantas | Válido, descriptivo |
| Importe | Válido, descriptivo |
| A | Válido pero no aporta información del contenido o función |
| AMC12 | Válido |
| AM12C | Válido |
| Coches usados | No válido (incluye un espacio) |
| Cochesusados ó CU | No válido (una variable tiene un único nombre) |
| Coches>30CV | No válido (incluye símbolo >) |
| Probabilidaddeaccidenteenbasealosdatosconocidos | Válido, pero no recomendable por ser excesivamente largo |
| Coches,motos | No válido (incluye una coma) |

Contenido y asignación de contenido a variables

El contenido de una variable puede ser numérico o alfanumérico. La existencia de texto mezclado con números, independientemente de si estos están delante, en medio o detrás del texto, se considerará equivalente a texto.

| | | |
|-----------------|---|----------------------|
| 123 | → | Numérico |
| -3233,75 | → | Numérico |
| Hay que mejorar | → | Alfanumérico (texto) |
| 123 coches | → | Alfanumérico (texto) |
| Son 35,37 € | → | Alfanumérico (texto) |

Cada variable tiene un único contenido en un momento dado, al que llamamos valor actual. Dicho valor puede cambiar en el tiempo pero

siempre es único, no quedando registro o memoria de cuáles fueron los contenidos anteriores de la variable. Puede haber dos variables o más con igual contenido.

La asignación de contenido a una variable se hará escribiendo el nombre seguido de un signo igual y del contenido a asignar, entrecomillado en el caso de que sea texto.

Así:

Salario = 30500

Salario = "Se asignarán 30500"

Una variable puede adoptar el contenido de otra variable. También puede adoptar el valor resultado de operaciones matemáticas entre números o entre distintas variables, siempre que sean variables numéricas.

Así:

| | | |
|--|---|--|
| A = 5 + 2 B = 32 Suma = A + B | → | Es válido. Suma adopta el valor 39. |
| A = "5 coches" B = "32 coches" Suma = A + B | → | No es válido. |
| A = 5 B = 32 C = A * B D = A + C E = D | → | Es válido. C toma el valor 160. D toma el valor 165 E toma el valor 165 |

La asignación del contenido de una variable a otra puede originar confusión por no saber qué variable adopta el valor de la otra. Se considerará siempre que el término a la izquierda de la expresión adopta el contenido del término a la derecha. Así $E = D$ significa "E adopta el valor de D" y, por supuesto, D no ve alterado su valor. Algunos programadores escriben pseudocódigo utilizando $E \leftarrow D$. En lugar de $E = D$, indicando así que "E toma el valor de D".

Otra cuestión que puede resultar problemática en cuanto a las variables es su contenido inicial. En algunos lenguajes si una variable es invocada y no se le ha asignado valor, se considerará que tiene valor cero (o cadena vacía si es de tipo texto), mientras que en otros da lugar a error o a valor no definido.

Así, si no se ha definido la variable B: $A = B * 7$ puede dar lugar a:

- a) La variable A toma el valor cero porque B, por defecto, vale cero.
- b) Error o valor no definido por no asignarse a B un valor de defecto.

Este último apartado sería resuelto haciendo:

$$\begin{aligned} B &= 0 \\ A &= B * 7 \end{aligned}$$

En la escritura de pseudocódigo nos será indistinto definir o no el contenido nulo de las variables. Sí será ineludible tenerlo en cuenta cuando el lenguaje informático que empleemos así lo requiera.

1 Variables booleanas

Con el fin de facilitar la escritura y razonamientos de programas se admite el uso de variables bipolares o que sólo admiten dos valores: verdadero (true) o falso (false).

La asignación de contenido se hace tomando la variable igual a uno de los dos valores.

Ejemplo:

Agotamiento = Verdadero
Optimizado = Falso

No hacen falta comillas porque no existe tratamiento como texto al tratarse de palabras clave. En realidad, el ordenador las tratará como variables binarias y adoptan valores numéricos (0,1 ó 0,-1) pero ni que decir tiene que nunca deberemos operar matemáticamente con este tipo de variables. El valor inicial para una variable booleana en caso de que no se haya definido lo consideraremos como *falso* o *cero*, con las reservas que hemos expuesto anteriormente.

No admitiremos que una variable que haya sido tipo booleana pase a ser de otro tipo.

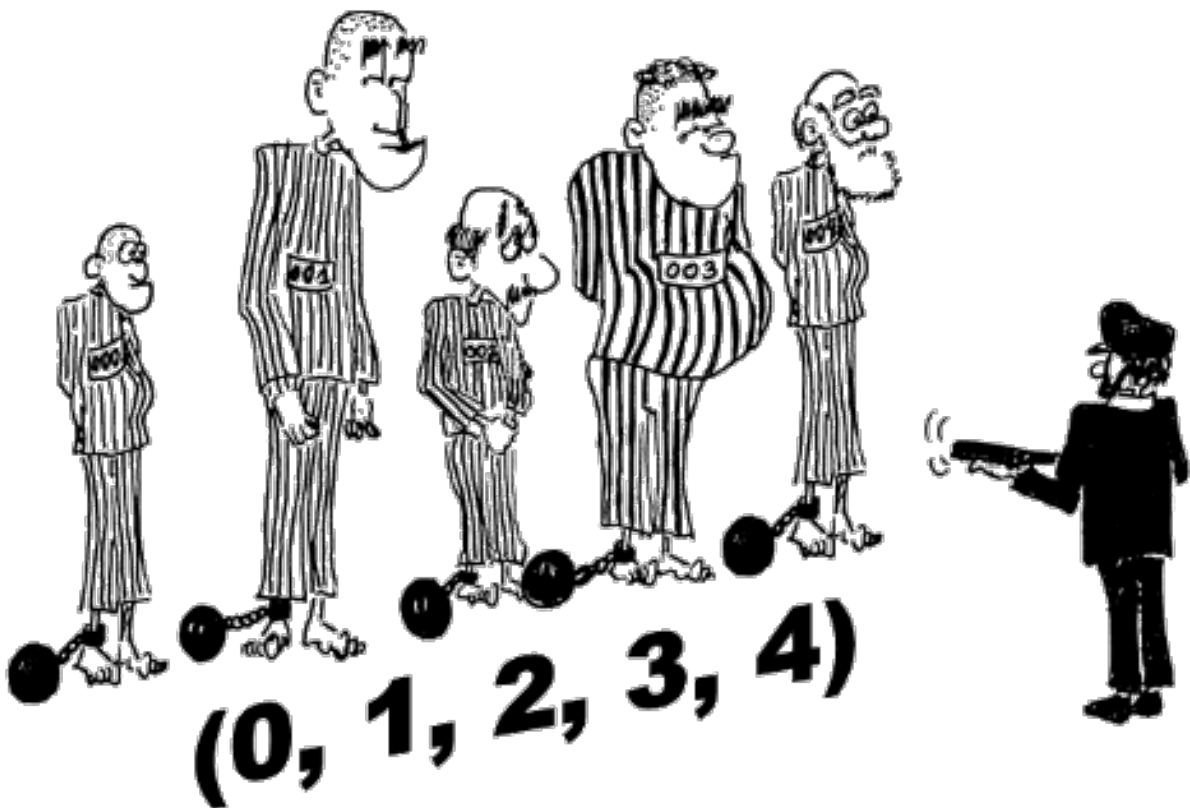
Ejemplo:

Agotamiento = Verdadero
Optimizado = 553
¡ESTO NO ES VÁLIDO!

En general, evitaremos modificar el tipo de información que contiene una variable. Si una variable, por ejemplo, la creamos como numérica la mantendremos siempre como tal.

1 Variables con índice o localizador. Arreglos

Se llama array (también se usan los términos matriz, formación o arreglo) a una serie de variables que tienen el mismo nombre pero que se diferencian por un número entero al que llamamos índice o localizador. Dicho número identifica a cada variable del array sin formar parte en sentido estricto del nombre de la variable: por ello el número puede ser sustituido por una operación, otra variable, o una operación entre números y variables.



1 Las variables como base de cálculo

Hay un aspecto diferenciador de la forma de calcular de un humano y la de un ordenador. Los humanos podemos guardar datos en espacios de memoria temporales que no tienen asignado un nombre mientras que el ordenador sólo puede guardar datos a través de variables que necesariamente tienen un nombre. Un humano puede operar de la siguiente manera:

“7 por 2 son 14; si lo multiplico por 6 son 84 y si lo divido por 2 son 42;
le sumo 4 y son 46, que dividido entre 2 son 23”

Un ordenador tendrá que operar a través de variables, por ejemplo:

a) $A = 7 * 2$; $A = A * 6$; $A = A / 2$; $A = A + 4$; $A = A / 2$

b) $A = 7 * 2$; $B = A * 6$; $C = B / 2$; $D = C + 4$; $E = D / 2$

En el caso a) el valor actual de la variable A es 23, habiéndose perdido todos los valores intermedios que fue tomando. En el caso b) el valor final de la serie de cálculos está registrado en la variable E, mientras que las variables A, B, C, D contienen los resultados intermedios.

La cantidad de variables a emplear en un proceso de cálculo dependerá del criterio del programador, quien habrá de valorar el interés que puede tener el conservar resultados intermedios. Con los criterios de economía y eficiencia, buscaremos siempre utilizar el menor número de variables posibles.