

Algoritmos Lineales

Un algoritmo es la descripción unívoca y finita de la secuencia de acciones a ejecutar para resolver un problema.

Esta definición merece ser analizada detenidamente.

'*es la descripción...de la secuencia de acciones*': vale decir que el algoritmo no hace, sino que describe (relata, cuenta) lo que se debe hacer.

'*unívoca*': esto significa que luego de ejecutada una determinada acción, la siguiente está indicada sin ambigüedades. Si después de ejecutada una acción existen dos o más que podrían ejecutarse y no existe un criterio para seleccionar la que corresponde, no es un algoritmo.

'*finita*': se refiere a que la secuencia de acciones debe finalizar en algún momento, cuando el problema esté resuelto. Una secuencia de acciones que podría llegar a ejecutarse indefinidamente, no es un algoritmo.

'*para resolver un problema*': o sea que antes de desarrollar un algoritmo, debemos conocer el problema que se quiere resolver, es decir, debemos haber hecho la especificación del mismo.

Los algoritmos pueden expresarse de distintas maneras según quién vaya a ejecutar las acciones: una guía de costura, una receta de cocina, una hoja de ruta, son algoritmos desarrollados utilizando lenguajes escritos y/o gráficos orientados a las personas que los ejecutarán.

Un algoritmo que será ejecutado en una computadora deberá ser escrito en un lenguaje de programación, pero en la etapa de desarrollo se dispone de dos alternativas: el pseudocódigo y los PSD (program structure diagram) también llamados estructogramas.

El **pseudocódigo** (pseudo: que parece) es un subconjunto del lenguaje natural que utiliza muy pocas palabras, cada una con un significado único, y una sintaxis estricta seimilar a los lenguajes de programación estructurados:

| <i>Palabra</i> | <i>Significado</i> | <i>Pascal</i> | <i>C</i> |
|--------------------|--|------------------|----------|
| Comienzo | Delimitador que indica que se inicia una secuencia de acciones | Begin | { |
| Fin | Delimitador que indica que finaliza una secuencia de acciones. | End | } |
| Leer (<ident>) | Acción de obtener el valor de una variable (magnitud simbolizada con un identificador) | Read Readln | getf |
| Escribir (<ident>) | Acción de enviar a la salida el valor de una variable. | Write Writeln | printf |

<ident> se reemplaza por el identificador. Ej.: Leer(x);

Existen también algunos símbolos especiales:

| <i>Símbolo</i> | <i>Uso</i> | <i>Pascal</i> | <i>C</i> |
|------------------|---|------------------------------------|---------------------------------------|
| ; | Indica la finalización de una acción. | Separa una acción de la siguiente. | Indica la finalización de una acción. |
| <ident> ← <expr> | Asigna a la variable de la izquierda el valor de la expresión que aparece a la derecha. | <ident> := <expr> | <ident> = <expr> |
| . | En Pascal se coloca después del último end para indicar el fin del código. | | |

Ejemplo: El problema SupTrian consiste en el cálculo de la superficie de un triángulo.

| <i>Especificación</i> | <i>Algoritmo</i> |
|--|--|
| <ol style="list-style-type: none"> SupTrian. Calcular la superficie de un triángulo. b(base), h(altura): n^{os} reales. s(superficie): n^o real. $s = b \cdot h / 2$ | Comienzo leer(b); leer(h); $s \leftarrow b \cdot h / 2$; escribir(s); Fin. |

El PSD (program structure diagram), es un modo gráfico de representar un algoritmo. En él, cada acción simple se representa como un rectángulo; una secuencia de acciones será a su vez un rectángulo formado por otros menores (las acciones compuestas que veremos más adelante tienen otras representaciones).

| |
|------------------------------|
| leer(b) |
| leer(h) |
| $s \leftarrow b \cdot h / 2$ |
| escribir(s) |

Parentemente, el PSD no agrega nada al pseudocódigo, pero responde a una norma internacional (ISO/IEC ...) por lo que pueden ser interpretados por cualquier persona en cualquier lugar del mundo, la cual podrá codificarlo en cualquier lenguaje de programación.

| <i>Programa en Pascal</i> | <i>Programa en C</i> |
|---|--|
| Program SupTrian; var b, h, s: real; begin readln(b); readln(h); s := b * h / 2; write(s); end. | <pre>#include <stdio.h> main() { float b, h, s; s = b * h / 2; scanf("%e", b); scanf("%e", h); printf("%e\n", s); }</pre> |

Puede verse que en ambos programas aparecen los puntos 2 y 3 de la especificación (en Pascal **b, h, s: real**; y en C **float b, h, s;**) indicando en ambos casos, y de diferente modo, que los identificadores b, h y s representan números reales.

También puede verse que el bloque ejecutable en Pascal es prácticamente la traducción al inglés del pseudocódigo.

NOTA: Los códigos en Pascal y C se muestran a modo de ejemplo y no forman parte del presente curso.

Queda claro que para poder escribir el algoritmo, debemos primero realizar la especificación del problema y para codificarlo en un lenguaje de programación, debemos tener a la vista la especificación y el algoritmo (en pseudocódigo o PSD).

¿Y qué sucede si hemos dividido el problema en problemas menores y como consecuencia tenemos varias especificaciones? En ese caso tenemos dos opciones: escribir un solo algoritmo con las acciones necesarias para resolver cada uno de los problemas o escribir un algoritmo para cada problema. Esta última alternativa es la más recomendable, ya que permitirá escribir programas modulares con porciones (procedimientos/funciones) que podrán ser reutilizadas en otros programas.

Ejemplo: El problema CantBald permite calcular la cantidad de baldosas necesaria para cubrir un patio rectangular. Para su resolución es necesario resolver antes los problemas SupRect para calcular la superficie del patio y SupCuad para calcular la superficie de una baldosa.

| <i>Especificaciones</i> | | |
|--|---|---|
| 1. SupRect. 2. Calcular la superficie de un rectángulo. 3. b(base), h(altura): n ^{os} reales. 4. s(superficie): n ^o real. 5. $s = b \cdot h$ | 1. SupCuad. 2. Calcular la superficie de un cuadrado. 3. l(lado): n ^o real; 4. s(superficie): n ^o real; 5 $s = l \cdot l$ | 1. CantBald. 2. Calcular la cantidad de baldosas. 3. Sp(sup. patio), Sb(sup. baldosa): n ^{os} reales. 4. Cb(cant. de baldosas): n ^o entero. 5. $Cb = Sp/Sb$. |

| <i>Algoritmos</i> | | |
|--|--|--|
| Comienzo leer(b); leer(h); $s \leftarrow b \cdot h$; escribir(s); Fin. | Comienzo leer(l); $s \leftarrow l \cdot l$; escribir(s); Fin. | Comienzo leer(Sp); leer(Sb); $Cb \leftarrow Sp / Sb$; escribir(Cb); Fin. |