4. Desarrollo de algoritmos

Un problema se puede resolver de diferentes maneras.

4.1. Preliminares

- ¿Qué es un algoritmo?

  Es un conjunto ordenado y finito de operaciones (instrucciones) que permite hallar una solución de un problema.

- ¿Por qué es importante?

  Porque representa el know-how.

- ¿Qué términos similares tenemos?

  Tenemos: receta, proceso, método, técnica, procedimiento, rutina; pero algoritmo denota un concepto riguroso, hasta donde es posible.

Como ejemplo, volvamos a ver el algoritmo de Euclides para calcular el MCD de dos enteros positivos.

```
Descripcion: Este algoritmo asume que la entrada consiste en dos enteros
positivos y se procede a calcular el MCD de estos dos valores.

Procedimiento:
Paso 1. Asignar M y N el valor del mayor y menor de estos dos valores.
Paso 2. Dividir M por N y llamar al resto de R.
Paso 3. Si R no es cero, entonces asignar a M el valor de N, asignar a
        N el valor de R y retornar al paso 2; caso contrario, el MCD es
        el valor actual de N.
```

A continuación, probemos que el algoritmo resuelve el problema de calcular el MCD de dos enteros positivos. En primer lugar definamos como nuestro universo el conjunto de los enteros positivos P. Dados $d, n \in P$, diremos que d divide a n, lo que denotaremos $d|n$, si $n = cd$, para algún $c \in P$. Luego, para todo $d, n, m \in P$ tenemos las siguientes propiedades:

- $d|n \quad \Rightarrow \quad d|nq.$
- $d|n,\ d|m \quad \Rightarrow \quad d|(n+m).$
- $d|n,\ d|m,\ n > m \quad \Rightarrow \quad d|(n-m).$

- $r = m \mod n \quad \Rightarrow \quad r < n.$

Ahora, vayamos con la prueba en sí del algoritmo. Si $R = 0$, entonces $mcd(M, N) = N$; caso contrario, probaremos que el conjunto de divisores de M y N es el mismo que el de N y R para luego afirmar que

$$\text{mcd}(M, N) = \text{mcd}(N, R) \quad \text{y} \quad R < N. \tag{4.1}$$

En efecto, bastaría probar

$$d|M,\ d|N \quad \Rightarrow \quad d|R \qquad \text{y}$$

$$d|N,\ d|R \quad \Rightarrow \quad d|M,$$

lo cual se sigue de las propiedades dadas arriba. Luego, sea $R_1 = N \mod R$. Si $R_1 = 0$, entonces $\text{mcd}(N, R) = R$, y por 4.1 concluimos que $\text{mcd}(M, N) = R$; caso contrario, análogamente como se procedió arriba, tenemos que

$$\text{mcd}(N, R) = \text{mcd}(R, R1) \quad \text{y} \quad R1 < R. \tag{4.2}$$

Luego, sea $R2 = R \mod R1$. Si $R_2 = 0$, entonces $\text{mcd}(R, R1) = R1$, y por 4.1 y 4.2 concluimos que $\text{mcd}(M, N) = R1$; caso contrario, tenemos que

$$\text{mcd}(R, R1) = \text{mcd}(R1, R2) \quad \text{y} \quad R2 < R1. \tag{4.3}$$

Por un argumento inductivo, este procedimiento debe terminar en algún momento, ya que, de 4.1, 4.2, 4.3,... tenemos

$$\ldots R2 < R1 < R < N.$$

4.2. Fases del desarrollo de un algoritmo

Las fases del desarrollo de un algoritmo son

- Análisis
- Diseño
- Cálculo

Veamos cada una de ellas:

4.2.1. Análisis

Estudia los límites de un problema:

- Salida: información requerida.
- Entrada: datos.

Por ejemplo, para calcular el MCD de dos enteros positivos una tabla ayuda a comprender el problema.

4.2.2. Diseño

Propone uno o más modelos de solución para el problema que se ocupa. Pasaremos por las siguientes etapas:

- **Modelo del algoritmo**: para nuestro ejemplo podemos plantear tres alternativas.
  1. Descomponer M y N en sus factores primos. Luego, el MCD es el producto de los factores primos comunes.
  2. Calcular el MCD directamente.
  3. Algortimo de Euclides.
- **Prueba del algoritmo**: hay diseños que requieren una prueba lógica, como para el del algoritmo de Euclides; pero en los dos primeros casos se ve claramente que el algoritmo aplica la definición de MCD directamente.
- **Representación del algoritmo**: luego de encontrar un algoritmo, es necesario explicarlo a otras personas; para el mismo solucionador, las ideas no son completamente claras, es importante representar la solución en diversas formas. Nosotros utilizaremos los diagramas de flujos y los pseudocódigos.

4.2.3. Cálculo

Los algoritmos suelen requerir muchos cálculos, y se hace necesario calcular los resultados con la ayuda de una computadora. En esta fase, utilizaremos scratch para nuestros cálculos, luego usaremos un lenguaje de programación.

4.3. Partes de un algoritmo

Operativamente un algoritmo suele descomponerse en tres partes:

- **Inicio**: condiciones de entrada u otras instrucciones.
- **Proceso**: instrucciones.
- **Fin**: condiciones de salida u otras instrucciones.

Ejercicio 4.1. Realice el desarrollo de un algoritmo para calcular el producto de dos número complejos.

Tarea: Leer páginas 272-279 de [Brookshear and Brylow, 2015].

Referencias

[Brookshear and Brylow, 2015] Brookshear, G. and Brylow, D. (2015). *Computer Science - An Overview*. Pearson Education Limited, 12th edition.