

TEMA 0: INTRODUCCIÓN

ALGORITMOS Y ESTRUCTURAS DE DATOS

M. Colebrook Santamaría

J. Riera Ledesma

J. Hernández Aceituno

Objetivos

- ¿Qué es un Algoritmo?
- Definición formal de algoritmo
- Ejemplos de Algoritmo
- Representación de un Algoritmo

¿Qué es un algoritmo? (1)

Según el DRAE (Dicc. de la Real Acad. Esp.) se define:

- **algoritmo:** (del griego *arithmos*, “número”)
 1. Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.
 2. Método y notación en las distintas formas del cálculo.
- **algoritmia:** Ciencia del cálculo aritmético y algebraico (i.e.: estudio de los algoritmos); teoría de los números.
- **algorítmico:** Perteneciente o relativo a los algoritmos.

Ejemplos: sumar, restar, multiplicar, recetas de cocina, etc.

¿Qué es un algoritmo? (2)

- La ejecución de un algoritmo *normalmente* **no** debe implicar:
 - Ninguna decisión **subjetiva**.
 - El uso de la **intuición** o la **creatividad**.
 - Ninguna noción **vaga** (sin detalles cuantificables).
- Una excepción son los algoritmos probabilísticos, que efectúan elecciones aleatorias y cuya probabilidad es conocida y está controlada.
- Por otra parte, existen problemas para los cuales no existen actualmente algoritmos prácticos (que calculen la solución en un tiempo razonable).

¿Qué es un algoritmo? (3)

- Definición formal: Un **algoritmo** es un conjunto de **instrucciones** o reglas bien **definidas**, **ordenadas** y **finitas** que permite realizar una actividad mediante pasos sucesivos sin **ambigüedad**.
- Algunos ejemplos de algoritmos podrían ser:
 - Manuales de usuario/instrucciones
 - Recetas de cocina
 - Aritmética elemental: multiplicar, potencias, raíces...
 - Máximo común divisor (mcd)
 - Mínimo común múltiplo (mcm)
 - Cambio de base de un número

Representación de un Algoritmo (1)

Los algoritmos pueden ser expresados de diferentes formas: lenguaje natural, pseudocódigo, diagramas de flujo, lenguajes de programación... El diseño de un algoritmo usualmente se hace en 3 niveles:

1. **Descripción de alto nivel:** definir el problema a resolver, determinar la solución que se desea obtener (como un modelo matemático) y explicar el algoritmo verbalmente a grandes rasgos.
2. **Descripción formal:** describir la secuencia de pasos que encuentran la solución mediante pseudocódigo.
3. **Implementación:** expresar el algoritmo en un lenguaje de programación específico.

Representación de un Algoritmo (2)

- Descripción de alto nivel:

“Dado un número real positivo x , queremos encontrar su raíz cuadrada r por tanteo.

Daremos a r un cierto valor inicial (digamos 0) y la iremos ajustando hasta que la diferencia entre r^2 y x sea menor que un cierto margen (digamos 10^{-6}).

Para ajustar r , le sumaremos iterativamente un cierto valor real v (digamos 1), hasta que r^2 sea mayor que x .

Cuando ocurra esto, multiplicamos v por $-1/2$ para ajustar en la otra dirección. Repetimos hasta que encontremos el valor correcto de r o hasta que $|v|$ sea demasiado pequeño (digamos menor que 10^{-6}).”

Representación de un Algoritmo (3)

■ Descripción de alto nivel:

Por ejemplo, si $x = 5$:

- $v = 1$ $r = 0 \rightarrow 1 \rightarrow 2 \rightarrow 3$, parar ($3^2 > 5$)
- $v = -1/2$ $r = 3 \rightarrow 2.5 \rightarrow 2$, parar ($2^2 < 5$)
- $v = 1/4$ $r = 2 \rightarrow 2.25$, parar ($2.25^2 > 5$)
- $v = -1/8$ $r = 2.25 \rightarrow 2.125$, parar ($2.125^2 < 5$)
- ...
- $v = -2^{-19}$ $r = 2.236067 \rightarrow \dots \rightarrow 2.236068$, parar
 $|r^2 - x| < 10^{-6} \rightarrow \sqrt{5} \approx 2.236068$

Representación de un Algoritmo (4)

- Descripción formal:

sea $x > 0$, $v = 1$, $r = 0$, $\varepsilon = 10^{-6}$

mientras que $|r^2 - x| > \varepsilon$ *y* $|v| > \varepsilon$, *hacer*

$r = r + v$

si $(v > 0$ *y* $r^2 > x)$ *o* $(v < 0$ *y* $r^2 < x)$,

entonces

$v = -\frac{1}{2} \cdot v$

Representación de un Algoritmo (5)

■ Implementación:

```
double raiz(const double x, double eps = 1e-6) {  
    double r = 0;  
    for (double v = 1; abs(r * r - x) > eps &&  
        abs(v) > eps; r += v)  
        if (((r * r > x) == (v > 0)) ||  
            ((r * r < x) == (v < 0)))  
            v *= -.5;  
    return r;  
}
```

Referencias

- ★ Brassard, G. and Bratley, P. (1998) “Fundamentos de Algoritmia”, Prentice-Hall. [**Capítulo 1**]
- ★ Concepto de Algoritmo en Wikipedia:
es.wikipedia.org/wiki/Algoritmo