Análisis y Diseño de Algoritmos

Algoritmo: Conjunto ordenado y finito de operaciones (instrucciones) que permite hallar la solución de un problema.

Ejemplo: hallar el máximo común divisor (mcd) de dos enteros m y n, mayores que 0. Utilicemos el algoritmo de Euclides:
Pasos:

1) Hallar el resto de r de dividir m/n
2) Si r = 0, terminar, n es el mcd.
3) Asignar: m $\leftarrow$ n, n$\leftarrow$ r, volver al paso 1.

Prueba de que el algoritmo (modelo) resuelve el problema
Después del paso 1, tenemos:

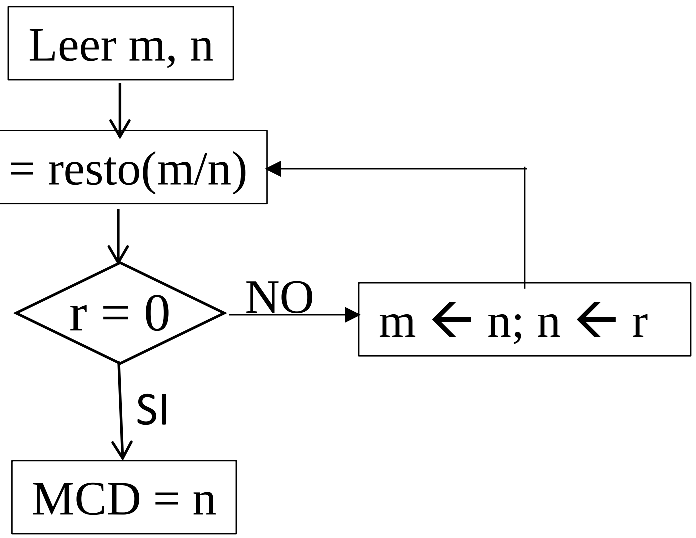
$m = qn + r$ para algún entero q,

Si r = 0, entonces m es un múltiplo de n y, en tal caso n es el mcd de m y n.
Si $r \neq 0$, cualquier número que divida a m y n, divide a $m - qn = r$, y
cualquier número que divida a n y r divide a $qn + r = m$

Es decir que, el conjunto de divisores de m, n es el mismo que el conjunto de divisores de n y r; en particular, el mcd de m, n es el mismo que el mcd de n, r. Por lo tanto el paso 3 no cambia la respuesta al problema original.

Podemos representar el algoritmo en forma gráfica:

Diagrama



También podemos representar el algoritmo en un lenguaje de programación.

Análisis y Diseño de Algoritmos

Análisis: Estudio, mediante técnicas informáticas, de los límites, características y posibles soluciones de un problema al que se aplica un tratamiento de computación.

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

Importancia: Es muy importante, pues representa **el saber hacer, el know how**.

Diseño

Descripción o bosquejo del algoritmo. Lo podemos dividir en 3 grandes componentes:

Inicio : 0 o más condiciones de **entrada** y 0 o más instrucciones.
Proceso : 0 o más instrucciones
Fin : 1 condición de **salida** y 0 o más instrucciones de fin

Se suelen seguir una de las dos secuencias:

Inicio → Proceso → Fin
Inicio → Fin → Proceso

Ejemplo: Algoritmo para multiplicar números complejos
Multiplicación de dos números: (a + ib) * (c + id) = ac - bd + i(ad + bc))

Análisis

1) Límites:
   a, b, c y d son números reales,
   -32768 < a, b, c, d < -0.0000001
   0.0000001 < a, b, c, d < 32767
2) Herramientas: El algoritmo será desarrollado en C
3) Posibles soluciones: una sola:
   (a + ib) * (c + id) = ac - bd + i(ad + bc))

Diseño (Algoritmo)

Inicio:

Leer los valores a: a, b, c y d
Definir e y f, para los resultados finales

Proceso:

e = ac - bd
f = ad + bc

Fin:

Escribir el resultado: e + if
Controlar la calidad del inicio y el proceso.

Programa: Programar en C, está fuera del alcance de esta demostración.

Ejemplo: **Análisis** **y** **Diseño** **de un** Algoritmo para una tabla de grados Fahrenheit a Centígrados de 0 a 50 grados Farenheit de 2 en 2.

Objeto a tratar

Temperatura medida en grados fahrenheit (F) y centígrados (C)

Multiplicación de dos números: C = (F – 32) / 1.8

Análisis

1) Límites:

   F = 0, 2, 4, … 50

2) Características:

   El algoritmo será desarrollado en C

3) Posibles soluciones: una sola:

   C = (F – 32) / 1.8

Diseño (pseudo código en C)

4) Descripción de algoritmo: Inicio, Proceso y Fin

```
Dar valores: inicio = 0, fin = 50
F = inicio
Escribir el título de la tabla
Mientras (F < fin) {
        C = (F – 32) / 1.8
        Imprimir F y C
        C = C + 2
}
```

Análisis y Diseño de Algoritmos para Ciencias de la Computación

Estudio, mediante técnicas computación, de:

1) Los límites
2) Las características
    1) **Finitud**: Tener un número finito de pasos
    2) **Definibilidad**: Cada paso del algoritmo debe definirse en modo preciso, ejemplo el Primer paso del algoritmo de Euclides tiene una ambigüedad: Se requiere elegir como m el mayor de los números
    3) **Entrada**: Un *algoritmo tiene cero o más entradas*
    4) ***Salida**: Un algoritmo tiene* una o más salidas
    5) **Efectividad**: Todas las operaciones deben ser básicas para poderse ejecutar en modo exacto y en un tiempo finito por una persona.
3) Posibles soluciones que interpretan el problema a resolver
4) Descripción o bosquejo de la solución elegida

    Ejemplo: hallar el máximo común divisor (mcd) de dos enteros m y n, mayores que 0.

    Utilicemos el algoritmo de Euclides (E):

Formalización de la definición de algoritmo

<u>Notación</u>:

Identificar al algoritmo en forma corta:

E: Hallar el máximo común divisor (mcd) de dos enteros m y n.

Identificación de los pasos: E1, E2, …

Indicadores de operadores:

Asignación: $\leftarrow$ relación: $<, =, >$

Resumen del paso, entre corchetes:

E0: [elegir el mayor] m $\leftarrow$ mayor, n $\leftarrow$ menor

Descripción integrada de E mediante la cuaterna: (Q, I, Ω, f), donde:

- I : Entradas
- Ω : Salidas
- Q : Estados de cálculo, incluye a I y Ω
- f : $Q \rightarrow Q$. **<u>La función f es el modelo de solución</u>.**

Cada entrada x en I define una secuencia de pasos: $x_0 \leftarrow x$, $x_{k+1} \leftarrow = f(x_k)$.

La secuencia de pasos acaba en k, si k es el menor entero para el cual x_k está en Ω

f es la función identidad sobre Ω: $x_{k+1} = f(x_k) = x_k$

Marca de fin de algoritmo: $\blacksquare$

<u>Definición de Algoritmo A</u>:

A = (I, Ω, Q, f**)**

Ejemplo: Definición del algoritmo de Euclides E, para hallar el MCD

I = {(m, n) / m y n son enteros > 0}
Ω = {(n) / n es entero > 0}
Q = Ω + I + {(m, n, r, 1), (m, n, r, 2), (m, n, p, 3) / r entero >= 0, p entero > 0}
f(m, n) ← (m, n, 0, 1)
f(n) ← (n) [acaba la próxima vez que se repite (n)]
f(m, n, r, 1) ← (m, n, resto de m/n, 2) [Hallar el resto]
f(m, n, r, 2) ← (n) si r = 0; (m, n, r, 3) en caso contrario. [Fija condición de fin]
f(m, n, p, 3) ← (n, p, p, 1) ∎ [Intercambio]

Definición del Algoritmo E:
E = (I, Ω, Q, f**)**

Todavía no se garantiza la condición de efectividad; para ello es necesario imponer restricciones en Q, I, Ω y f.