1. Algoritmia Elemental

La algoritmia es el estudio de los algoritmos: secuencias finitas de instrucciones bien definidas que permiten resolver problemas o realizar tareas. Antes de diseñar o analizar algoritmos, necesitamos dominar algunas herramientas matemáticas fundamentales.

1.1. Preliminares

Los preliminares son conceptos básicos necesarios para entender y trabajar con algoritmos de manera rigurosa.

1.1.1. Notación

La notación es el lenguaje simbólico que usamos para describir problemas y algoritmos. Algunos aspectos importantes incluyen:

- Notación de conjuntos:
  - $\mathbb{N}$: conjunto de los números naturales (0, 1, 2, 3, ...).
  - $\mathbb{Z}$: conjunto de los números enteros (..., -2, -1, 0, 1, 2, ...).
  - $\mathbb{R}$: conjunto de los números reales.
  - Ejemplo: $A = \{x \in \mathbb{N} \mid x \text{ es par}\}$.

- Cuantificadores lógicos:
  - $\forall$ (para todo): Expresa que una afirmación vale para todos los elementos de un conjunto.
  - $\exists$ (existe): Indica que hay al menos un elemento que satisface una condición.
  - Ejemplo: $\forall x \in \mathbb{N}, \exists y \in \mathbb{N}$ tal que $y = x + 1$.

- Relaciones y funciones:
  - Una relación es un conjunto de pares ordenados. Ejemplo: 'ser menor que'.
  - Una función asocia a cada elemento de un conjunto exactamente un elemento de otro conjunto.

- Notación asintótica:
  - $O(f(n))$: cota superior del crecimiento de un algoritmo.

- $\Omega(f(n))$: cota inferior.
- $\Theta(f(n))$: cota ajustada (crecimiento exacto).

- Sumatorias y productorias:
  - Suma de términos: $\Sigma_{i=1}^{n} i = 1 + 2 + ... + n$.
  - Producto de términos: $\Pi_{i=1}^{n} i = 1 \times 2 \times ... \times n = n!$ (factorial).

1.1.2. Contradicción

La prueba por contradicción es un método lógico muy usado en matemáticas y algoritmia.

Idea principal: Para probar que una afirmación P es verdadera:

1. Suponemos que P es falsa.
2. A partir de esa suposición, deducimos algo absurdo o imposible (una contradicción).
3. Por lo tanto, P debe ser verdadera.

Ejemplo clásico:

Probar que $\sqrt{2}$ no es un número racional.

- Se supone que $\sqrt{2} = p/q$ (con p y q enteros primos entre sí).
- Se llega a la contradicción de que ambos serían pares, lo cual contradice que sean primos relativos.

1.1.3. Inducción matemática

La inducción matemática es un método poderoso para demostrar que una propiedad P(n) es verdadera para todos los números naturales n.

Dos pasos fundamentales:

1. Base de inducción: Verificar que la propiedad es verdadera para el primer caso (generalmente $n=0$ o $n=1$).
2. Paso inductivo:
   - Supongamos que la propiedad es verdadera para $n = k$ (hipótesis inductiva).
   - Demostrar que entonces también es verdadera para $n = k+1$.

Ejemplo clásico:

Demostrar que:

$\Sigma_{i=1}^{n} i = n(n+1)/2$

- Base: Para n=1, 1 = (1(1+1))/2 = 1.
- Paso inductivo: Suponer que la fórmula es cierta para n=k.
- Se prueba para n=k+1:

$(1+2+3+...+k) + (k+1) = (k(k+1))/2 + (k+1) = ((k+1)(k+2))/2.$

1.1.4. Problemas

Problema 1 (Notación):

Escribe usando sumatoria el siguiente cálculo:

$1^2 + 2^2 + 3^2 + ... + 10^2$

Respuesta: $\Sigma_{i=1}^{10} i^2$.

```
Algoritmo SumaDeCuadrados

   Definir i, suma Como Entero

   suma <- 0

   Para i <- 1 Hasta 10 Con Paso 1 Hacer

       suma <- suma + (i * i)

   FinPara

   Escribir "La suma de los cuadrados de 1 a 10 es: ", suma

FinAlgoritmo
```

Problema 2 (Contradicción):

Prueba que no existen enteros impares cuya suma sea par e impar al mismo tiempo.

Idea de solución:

- Supón que existen tales enteros.
- Su suma debería ser par y a la vez impar, lo cual es una contradicción.

```
Algoritmo SumaDeImpares
  Definir a, b, suma Como Entero
  Escribir "Ingrese el primer número impar:"
  Leer a
  Escribir "Ingrese el segundo número impar:"
  Leer b
  Si (a MOD 2 = 1) Y (b MOD 2 = 1) Entonces
    suma <- a + b
    Si suma MOD 2 = 0 Entonces
      Escribir "La suma de ", a, " y ", b, " es ", suma, ", que es par."
    Sino
      Escribir "Error: la suma debería ser par, pero es impar."
    FinSi
  Sino
    Escribir "Ambos números deben ser impares."
  FinSi
FinAlgoritmo
```