Universidad Diego Portales

Apunte Clase 4: Números enteros

Yerko Ortiz
Rodrigo Alvarez
2024

Contents

1 Números enteros en la computación

Los números enteros son piezas fundamentales en la computación y tienen un impacto profundo en el diseño y funcionamiento de algoritmos. Como la unidad básica de información, los números se utilizan en casi todos los aspectos de la informática, desde operaciones matemáticas simples hasta complejos algoritmos criptográficos.

2 Divisibilidad

En matemáticas, la divisibilidad es la propiedad de un número de ser divisible entre otro número sin dejar un residuo. El número entero b divide al entero a si y solo sí $a = b \cdot k$ para algún entero k. En la nomenclatura también es posible decir que a es divisible por b o que b es *divisor* de a y escribimos esto como $b \mid a$. Si b no divide a a, escribimos $b \nmid a$.
Por ejemplo:

- 10 es divisible por 5 ($5 \mid 10$) porque $10 \div 5 = 2$ con un residuo de 0.
- 23 es divisible por 1 ($1 \mid 23$) porque $23 \div 1 = 23$ con un residuo de 0.
- 9 no es divisible por 7 ($7 \nmid 9$) porque $9 \div 7 = 1$ con un residuo de 2.

Para validar si un número entero es divisor de otros se hace verificando si el *residuo* de la división es cero.
Este concepto es central y tiene una aplicación directa en muchos problemas algorítmicos:

2.1 Algoritmo de Euclides para el Máximo Común Divisor

Uno de los algoritmos más antiguos y eficientes relacionados con la divisibilidad es el algoritmo de Euclides, utilizado para calcular el Máximo Común Divisor (MCD o GCD en inglés) de dos números enteros. El MCD de dos números es el número entero más grande que es divisor de ambos números, es decir que divide a ambos sin dejar residuo.

El algoritmo de Euclides se basa en el principio de que el máximo común divisor de dos números no cambia si el número más grande se reemplaza por su diferencia con el número más pequeño. Por ejemplo 7 es el MCD de 49 y 35 (ya que $49 = 7 \times 7$ y $35 = 7 \times 5$), y el mismo 7 también es el MCD de 14 ($49 - 35$) y 35. Como este reemplazo reduce el número más grande, podemos repetir este proceso para obtener pares cada vez más pequeños sucesivamente hasta que el resto sea 0.

Para dos números a y b, donde $a > b$, este algoritmo sigue los siguientes pasos:

1. **Cálculo del residuo:** Divide a por b y calcula el residuo r, es decir $r = a \bmod b$

2. **Iteración:**

   - SI $r \neq 0$, se reemplaza a con b y b con r, y se repite el proceso.

3. **Terminar:** El proceso continúa hasta que el residuo sea 0. En ese punto, el MCD será el valor actual de b.

Ejemplo:
Consideremos los números 56 y 42.

$$a = 56,\ b = 42.$$

Calculamos $r = 56 \bmod 42 = 14$.

Como $r \neq 0$, continuamos.

Reemplazamos a por 42 y b por 14.

Calculamos $r = 42 \bmod 14 = 0$.

Ahora $r = 0$, por lo que el MCD es 14.

Así, el MCD entre 56 y 42 es 14.
Esta idea algorítmica puede ser implementada en java:

```
static int mcd(int a, int b) {
    int r = a % b;
    while (r != 0) {
        a = b;
        b = r;
        r = a % b;
    }
    return b;
}
```

3 Números Primos

Los números primos son elementos fundamentales en matemáticas y juegan un papel crucial en la teoría de números, algoritmos, y diversas aplicaciones prácticas, como la criptografía. Un número primo es un número entero mayor que 1 que solo tiene dos divisores: 1 y sí mismo. Todo número entero p que es solo divisible por 1 y sí mismo se dice que es un número primo.

- **Ejempos de números primos**:
  - 2 es primo porque solo se puede dividir por 1 y 2.
  - 3 es primo porque solo se puede dividir por 1 y 3.
  - 5 es primo porque solo se puede dividir por 1 y 5.

- **Ejemplos de números no primos**:
  - 4 no es primo porque puede dividirse por 1, 2 y 4.
  - 6 no es primo porque puede dividirse por 1, 2, 3 y 6.

La existencia de los números primos da píe al teorema fundamental de la aritmética, el cual denota que todo número entero puede ser expresado de una y solo una factorización de números primos.
Por ejemplo los factores primos de 105 son 3, 5, 7. Como así mismo los factores primos de 32 son 2, 2, 2, 2, 2.
Identificar si un número es primo es un problema clave, especialmente en campos como la criptografía. A continuación, se describe un método común para determinar si un número es primo o no.

3.1 Prueba de divisibilidad básica

Una manera sencilla de determinar si un número entero N es primo es comprobando si es o no divisible por algún número entre 2 y $\sqrt{N}$. No es necesario comprobar los mayores a $\sqrt{N}$ porque sus pares ya habrán aparecido con divisores menores. De esta manera si N es divisible por cualquier número en el rango no sería primo, y si se cumple el caso contrario y N no es divisible por ningún valor dentro de ese rango, significa que N es primo.

Ejemplo:

Para determinar si 29 es primo, debemos comprobar divisibilidad por números hasta $\sqrt{29}$ (≈ 5). Así:

$$29 \bmod 2 = 1,\ 29 \text{ no es divisible por } 2$$
$$29 \bmod 3 = 2,\ 29 \text{ no es divisible por } 3$$
$$29 \bmod 4 = 1,\ 29 \text{ no es divisible por } 4$$
$$29 \bmod 5 = 4,\ 29 \text{ no es divisible por } 5$$
$$29 \bmod 6 = 5,\ 29 \text{ no es divisible por } 6$$

Dado que 29 no es divisible por ninguno de estos, es un número primo.

```
public static boolean isPrime(int N) {
    for (int i = 2; i * i < N; i++) {
        if (N % i == 0) return false;
    }
    return true;
}
```

4 Problema: Factores Primos

Diseñe un algoritmo que recibe como entrada un entero N y retorna un arreglo de enteros correspondiente a los factores primos que componen a N.