Taller ProgComp: Track Matemática

GCD y LCM

Gabriel Carmona Tabja

Universidad Técnica Federico Santa María,
Università di Pisa

April 8, 2024

Part I

Greatest Common Divisor

GCD

Problema

Dado dos enteros no negativos a y b, queremos encontrar el número más grande que divide tanto a y b.

$$gcd(a, b) = max\{k > 0 : (k|a) \text{ and } (k|b)\}$$

Si uno de los números es 0, la respuesta es el número que no es 0.

Solución Inicial

```
int gcd(int x, int y) {
  if(x == 0) return y;
  if(y == 0) return x;

  int max_div = 0;
  for(int i = 1; i <= min(x, y); i++) {
    if(x % i == 0 && y % i == 0) {
      max_div = i;
    }
  }
  return max_div;
}
```

¿Cuál es la complejidad?

Solución Inicial

```
1  int gcd(int x, int y) {
2    if(x == 0) return y;
3    if(y == 0) return x;
4
5    int max_div = 0;
6    for(int i = 1; i <= min(x, y); i++) {
7      if(x % i == 0 && y % i == 0) {
8        max_div = i;
9      }
10   }
11   return max_div;
12 }
```

¿Cuál es la complejidad?
La complejidad sería $O(min(x, y))$

Solución Inicial

```
1  int gcd(int x, int y) {
2    if(x == 0) return y;
3    if(y == 0) return x;
4
5    int max_div = 0;
6    for(int i = 1; i <= min(x, y); i++) {
7      if(x % i == 0 && y % i == 0) {
8        max_div = i;
9      }
10   }
11   return max_div;
12 }
```

¿Cuál es la complejidad?
La complejidad sería $O(min(x, y))$
¿Será muy lento?

Solución Inicial

```
int gcd(int x, int y) {
  if(x == 0) return y;
  if(y == 0) return x;

  int max_div = 0;
  for(int i = 1; i <= min(x, y); i++) {
    if(x % i == 0 && y % i == 0) {
      max_div = i;
    }
  }
  return max_div;
}
```

¿Cuál es la complejidad?
La complejidad sería $O(min(x, y))$
¿Será muy lento?

- Si $min(x, y) = 10^5$, no es tan lento.
- Si $min(x, y) = 10^{18}$, :(.

EUCLIDEAN ALGORITHM

Idea original

Resta el número más chico al más grande hasta que uno de los dos sea 0.

¿Por qué?

- Si g divide a y b, también divide $a - b$.
- Si g divide $a - b$ y b, también divide $a = b + (a - b)$.

Por lo tanto el conjunto de divisores en común entre $\{a, b\}$ y $\{b, a - b\}$ coincide.

Una mejora

Si $a > b$, entonces $a > b$ hasta que se reste $b \lfloor \frac{a}{b} \rfloor$ veces.
Por lo que $a - b$ se puede reemplazar por $a - \lfloor \frac{a}{b} \rfloor b = a \bmod b$

Idea final

$$gcd(a, b) = \begin{cases} a, & \text{if } b = 0 \\ gcd(b, a \bmod b), & \text{otherwise.} \end{cases}$$

Algoritmo

```cpp
// implementacion recursiva
int gcd(int a, int b) {
  if (b == 0)
    return a;
  else
    return gcd(b, a % b);
}

// implementacion iterativa
int gcd(int a, int b) {
  while (b) {
    a %= b;
    swap(a, b);
  }
  return a;
}

// A partir C++17 gcd es parte de la libreria estandar
```

ALGORITMO

```cpp
// implementacion recursiva
int gcd(int a, int b) {
  if (b == 0)
    return a;
  else
    return gcd(b, a % b);
}

// implementacion iterativa
int gcd(int a, int b) {
  while (b) {
    a %= b;
    swap(a, b);
  }
  return a;
}

// A partir C++17 gcd es parte de la libreria estandar
```

¿Cuál es la complejidad?

ALGORITMO

```cpp
// implementacion recursiva
int gcd(int a, int b) {
  if (b == 0)
    return a;
  else
    return gcd(b, a % b);
}

// implementacion iterativa
int gcd(int a, int b) {
  while (b) {
    a %= b;
    swap(a, b);
  }
  return a;
}

// A partir C++17 gcd es parte de la libreria estandar
```

¿Cuál es la complejidad?

$O(\log min(a, b))$, ¿Por qué? La demostración se deja para el alumno.

Si alguien lo demuestra y me entrega un documento explicando correctamente, se ganará 1 punto a la resolución de problemas

Part II

Extended Euclidean Algorithm

Extended Euclidean Algorithm

Idea

$$gcd(a, b) = a \cdot x + b \cdot y$$

Gracias a la Bézout's Identity, se asegura que siempre se puede encontrar valores x e y.

Algoritmo

```cpp
// implementacion recursiva
int gcd(int a, int b, int& x, int& y) {
  if (b == 0) {
    x = 1;
    y = 0;
    return a;
  }
  int x1, y1;
  int d = gcd(b, a % b, x1, y1);
  x = y1;
  y = x1 - y1 * (a / b);
  return d;
}

// implementacion iterativa
int gcd(int a, int b, int& x, int& y) {
  x = 1, y = 0;
  int x1 = 0, y1 = 1, a1 = a, b1 = b;
  while (b1) {
    int q = a1 / b1;
    tie(x, x1) = make_tuple(x1, x - q * x1);
    tie(y, y1) = make_tuple(y1, y - q * y1);
    tie(a1, b1) = make_tuple(b1, a1 - q * b1);
  }
  return a1;
}
```

Part III

Least Common Multiple

LCM

Problema

Dado dos enteros positivos no negativos, encuentre el menor común múltiplo.

LCM

Problema

Dado dos enteros positivos no negativos, encuentre el menor común múltiplo.

Teorema

$$lcm(a, b) = \frac{a \cdot b}{gcd(a, b)}$$

DEMOSTRACIÓN

Tenemos $d = gcd(a, b) \Rightarrow \exists a', b' \in \mathbb{N}$ s.t. $a = a' \cdot d$ y $b = b' \cdot d$

DEMOSTRACIÓN

Tenemos $d = gcd(a, b) \Rightarrow \exists a', b' \in \mathbb{N}$ s.t. $a = a' \cdot d$ y $b = b' \cdot d$
Digamos $m = \frac{a \cdot b}{d}$, dado esto podemos decir que:

- $m = \frac{a \cdot b}{d} = \frac{a \cdot b' \cdot d}{d} = a \cdot b' \Rightarrow a | m$
- $m = \frac{a \cdot b}{d} = \frac{a' \cdot d \cdot b}{d} = a' \cdot b \Rightarrow b | m$

Ahora, sabemos que m es un múltiplo nos queda solo encontrar que m sea el más chico.

DEMOSTRACIÓN

Tenemos $d = gcd(a, b) \Rightarrow \exists a', b' \in \mathbb{N}$ s.t. $a = a' \cdot d$ y $b = b' \cdot d$
Digamos $m = \frac{a \cdot b}{d}$, dado esto podemos decir que:

- $m = \frac{a \cdot b}{d} = \frac{a \cdot b' \cdot d}{d} = a \cdot b' \Rightarrow a | m$
- $m = \frac{a \cdot b}{d} = \frac{a' \cdot d \cdot b}{d} = a' \cdot b \Rightarrow b | m$

Ahora, sabemos que m es un múltiplo nos queda solo encontrar que m sea el más chico.
Llamemos c como cualquier múltiplo de a y $b \Rightarrow \exists x, y \in$ s.t. $c = a \cdot x = b \cdot y$
Ahora, desarrollemos $\frac{c}{m}$.

DEMOSTRACIÓN

Tenemos $d = gcd(a, b) \Rightarrow \exists a', b' \in \mathbb{N}$ s.t. $a = a' \cdot d$ y $b = b' \cdot d$
Digamos $m = \frac{a \cdot b}{d}$, dado esto podemos decir que:

- $m = \frac{a \cdot b}{d} = \frac{a \cdot b' \cdot d}{d} = a \cdot b' \Rightarrow a | m$
- $m = \frac{a \cdot b}{d} = \frac{a' \cdot d \cdot b}{d} = a' \cdot b \Rightarrow b | m$

Ahora, sabemos que m es un múltiplo nos queda solo encontrar que m sea el más chico.
Llamemos c como cualquier múltiplo de a y $b \Rightarrow \exists x, y \in$ s.t. $c = a \cdot x = b \cdot y$
Ahora, desarrollemos $\frac{c}{m}$.

$$\frac{c}{m} = \frac{c \cdot d}{a \cdot b} = \frac{c(a \cdot s + b \cdot t)}{a \cdot b}, \text{ para algún } s, t \in \mathbb{Z}$$

$$= \frac{c \cdot a \cdot s}{a \cdot b} + \frac{c \cdot b \cdot t}{a \cdot b} = \frac{c \cdot s}{b} + \frac{c \cdot t}{a}$$

DEMOSTRACIÓN

Tenemos $d = gcd(a, b) \Rightarrow \exists a', b' \in \mathbb{N}$ s.t. $a = a' \cdot d$ y $b = b' \cdot d$
Digamos $m = \frac{a \cdot b}{d}$, dado esto podemos decir que:

- $m = \frac{a \cdot b}{d} = \frac{a \cdot b' \cdot d}{d} = a \cdot b' \Rightarrow a|m$
- $m = \frac{a \cdot b}{d} = \frac{a' \cdot d \cdot b}{d} = a' \cdot b \Rightarrow b|m$

Ahora, sabemos que m es un múltiplo nos queda solo encontrar que m sea el más chico.
Llamemos c como cualquier múltiplo de a y $b \Rightarrow \exists x, y \in$ s.t. $c = a \cdot x = b \cdot y$
Ahora, desarrollemos $\frac{c}{m}$.

$$\frac{c}{m} = \frac{c \cdot d}{a \cdot b} = \frac{c(a \cdot s + b \cdot t)}{a \cdot b}, \text{ para algún } s, t \in \mathbb{Z}$$

$$= \frac{c \cdot a \cdot s}{a \cdot b} + \frac{c \cdot b \cdot t}{a \cdot b} = \frac{c \cdot s}{b} + \frac{c \cdot t}{a}$$

Reemplazando c queda:

$$\frac{b \cdot y \cdot s}{b} + \frac{a \cdot x \cdot t}{a} = y \cdot s + x \cdot t \in \mathbb{Z}$$

.
Como $\frac{c}{m}$ es un resultado entero, además c es cualquier múltiplo de a y b, por lo tanto m debe ser el menor común múltiplo.

Código

```
int lcm (int a, int b) {
    return a / gcd(a, b) * b;
}
```

REFERENCES I