Algoritmos en teoría de números

Clase 22

IIC 1253

Prof. Cristian Riveros

Outline

Representación de números

Algoritmos clásicos

Outline

Representación de números

Algoritmos clásicos

Representación de números

Teorema

Sea $b > 1$. Si $n \in \mathbb{N} - \{0\}$, entonces se puede escribir de forma única como:

$$n \ = \ a_{k-1}b^{k-1} + a_{k-2}b^{k-2} + \ldots + a_1 b + a_0 \ = \ \sum_{i=0}^{k-1} a_i b^i$$

- $k \geq 1$,
- $a_0, \ldots, a_{k-1}$ menor que b $\quad (a_i < b) \quad$ y
- $a_{k-1} \neq 0$.

¿cuál es la representación de 123?

con $b = 10$: $\quad 123 \ = \ 1 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0$

con $b = 2$: $\quad 123 \ = \ 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$

con $b = 8$: $\quad 123 \ = \ 1 \cdot 8^2 + 7 \cdot 8^1 + 3 \cdot 8^0$

Representación de números

Demostración: por inducción fuerte

$$P(n) \;:=\; n = \sum_{i=0}^{k-1} a_i b^i$$

1. $P(1): \quad 1 = 1 \cdot b^0$ ✓

2. Suponemos que $P(n')$ se cumple para todo $n' < n$ y dem. $P(n)$:
   - Existe un único par $m, r \in \mathbb{Z}$ con $0 \leq r < b$ tal que: $n = m \cdot b + r$.
   - Como $m < n$ (¿por qué?), entonces por hipótesis de inducción:
   $$m = \sum_{i=0}^{k-1} a_i b^i \quad k \geq 1,\ a_i < b \text{ y } a_{k-1} \neq 0.$$
   - Reemplazando m tenemos que:
   $$n = \left(\sum_{i=0}^{k-1} a_i b^i \right) \cdot b + r = \sum_{i=0}^{k-1} a_i b^{i+1} + r$$
   - Definiendo $a'_0 = r$ y $a'_{i+1} = a_i$ para $0 \leq i \leq k$, tenemos que:
   $$n = a'_k b^k + a'_{k-1} b^{k-1} + \ldots + a'_1 b + a'_0$$
   con $k + 1 \geq 1$, $a'_i < b$ y $a'_k \neq 0$. $\square$

Representación de números

Teorema

Sea $b > 1$. Si $n \in \mathbb{N}$, entonces se puede escribir de forma única como:

$$n = a_{k-1} b^{k-1} + a_{k-2} b^{k-2} + \ldots + a_1 b + a_0 = \sum_{i=0}^{k-1} a_i b^i$$

- $k \geq 1$,
- $a_0, \ldots, a_{k-1}$ menor que b $\quad (a_i < b)$ $\quad$ y
- $a_{k-1} \neq 0$.

Desde ahora, decimos que la representación de ***n* en base *b*** es la secuencia:

$$(n)_b = a_{k-1} \ldots a_1 a_0$$

Ejemplo

$(123)_{10} = 123 \qquad (123)_2 = 1111011 \qquad (123)_8 = 173$

¿cómo encontramos la representación de n en base b?

Para $n \in \mathbb{N} - \{0\}$ y $b > 1$, deseamos encontrar los coeficientes $a_i < b$ tal que:

$$n \;=\; a_{k-1}b^{k-1} + \ldots + a_1 b + a_0$$

Sabemos que $n \;=\; q \cdot b + r$ para algún único par $q, r \in \mathbb{N}$ con $r < b$.

¿qué es q y r en la representación de n en base b?

Proposición

Para un $n \in \mathbb{N} - \{0\}$ y $b > 1$, si $(n)_b = a_{k-1} \ldots a_1 a_0$ y $n = q \cdot b + r$, entonces:

$$\begin{aligned} r &= a_0 \\ (q)_b &= a_{k-1} \ldots a_1 \end{aligned}$$

Demostración: ejercicio.

¿cómo encontramos la representación de n en base b?

Ejemplo

Para escribir 39 en base 2:

$$\begin{aligned} 39 &= 19 \cdot 2 + \mathbf{1} \\ 19 &= 9 \cdot 2 + \mathbf{1} \\ 9 &= 4 \cdot 2 + \mathbf{1} \\ 4 &= 2 \cdot 2 + \mathbf{0} \\ 2 &= 1 \cdot 2 + \mathbf{0} \\ 1 &= 0 \cdot 2 + \mathbf{1} \end{aligned}$$

Por lo tanto, $(39)_2 = 100111$.

Para escribir 39 en base 5:

$$\begin{aligned} 39 &= 7 \cdot 5 + \mathbf{4} \\ 7 &= 1 \cdot 5 + \mathbf{2} \\ 1 &= 0 \cdot 5 + \mathbf{1} \end{aligned}$$

Por lo tanto, $(39)_5 = 124$.

Algoritmo para conversión de base

Algoritmo

input : Número $n \in \mathbb{N} - \{0\}$, base $b \geq 2$
output: Una secuencia $(n)_b = a_{k-1} \ldots a_1 a_0$

```
Function ConversiónBase (n, b)
    q := n
    k := 0
    while q ≠ 0 do
        a_k := q mod b
        q := q div b
        k := k + 1
    return a_{k-1} ... a_1 a_0
```

¿cuál es el tiempo del algoritmo en términos de n?

¿cuál es el tamaño de $(n)_b$ con respecto a n?

Suponga que $|(n)_b| = k$.

- Como n tiene k dígitos en base b, entonces:

$$b^{k-1} \quad \leq \quad n \quad \leq \quad b^k - 1$$

- Despejando k, tenemos que:

$$\log_b(n+1) \quad \leq \quad k \quad \leq \quad \log_b(n) + 1$$

- Como k es un valor entero:

$$\lceil \log_b(n+1) \rceil \quad \leq \quad k \quad \leq \quad \lfloor \log_b(n) + 1 \rfloor$$

Teorema

Para todo $n \in \mathbb{N}$ y $b \geq 2$, se cumple que $|(n)_b| = \lceil \log_b(n+1) \rceil$.

Por lo tanto, $|(n)_b| \in \mathcal{O}(\log(n))$.

Representación y división de números

¿cómo sabemos que un número es divisible por 3?

$$\begin{aligned}
n \textbf{ mod } 3 &= \left(a_k \cdot 10^k + \ldots + a_1 \cdot 10 + a_0\right) \textbf{ mod } 3 \\
&= \Big((a_k \cdot 10^k) \textbf{ mod } 3 + \ldots + (a_1 \cdot 10) \textbf{ mod } 3 + a_0 \textbf{ mod } 3\Big) \textbf{ mod } 3 \\
&= \Big((a_k \cdot 1) \textbf{ mod } 3 + \ldots + (a_1 \cdot 1) \textbf{ mod } 3 + a_0 \textbf{ mod } 3\Big) \textbf{ mod } 3 \\
&= \Big(a_k + \ldots + a_1 + a_0\Big) \textbf{ mod } 3
\end{aligned}$$

Demuestre reglas para 4, 9, ...

Outline

Representación de números

Algoritmos clásicos

¿cómo sumamos dos números en base b?

Considere dos números en base 2 ($b = 2$):

$$\begin{aligned} n &= n_{k-1}2^{k-1} + \ldots + n_1 \cdot 2 + n_0 \\ m &= m_{k-1}2^{k-1} + \ldots + m_1 \cdot 2 + m_0 \end{aligned}$$

Sumando ambos números tenemos que:

$$n + m = (n_{k-1} + m_{k-1}) \cdot 2^{k-1} + \ldots + (n_1 + m_1) \cdot 2 + (n_0 + m_0)$$

¿estamos listos?

¿cómo sumamos dos números en base b?

Sumando ambos números tenemos que:

$$n + m \quad = \quad (n_{k-1} + m_{k-1}) \cdot 2^{k-1} + \ldots + (n_1 + m_1) \cdot 2 + (n_0 + m_0)$$

Para $(n_0 + m_0)$ sabemos que $(n_0 + m_0) = c_0 \cdot 2 + s_0$.

$$n + m \quad = \quad (n_{k-1} + m_{k-1}) \cdot 2^{k-1} + \ldots + (n_1 + m_1 + c_0) \cdot 2 + s_0$$

Para $(n_1 + m_1 + c_0)$ sabemos que $(n_1 + m_1 + c_0) = c_1 \cdot 2 + s_1$.

$$n + m \quad = \quad (n_{k-1} + m_{k-1}) \cdot 2^{k-1} + \ldots (n_2 + m_2 + c_1) \cdot 2^2 + s_1 \cdot 2 + s_0$$

$\ldots$

¿a qué corresponden los valores c_0, c_1, $\ldots$?

¿cómo sumamos dos números en base b?

Por lo tanto, se define recursivamente:

$$\begin{aligned}
n_0 + m_0 &= c_0 \cdot 2 + s_0 \\
n_1 + m_1 + c_0 &= c_1 \cdot 2 + s_1 \\
n_2 + m_2 + c_1 &= c_2 \cdot 2 + s_2 \\
&\ldots \\
n_{k-1} + m_{k-1} + c_{k-2} &= c_{k-1} \cdot 2 + s_{k-1}
\end{aligned}$$

Para lo cuál se obtiene:

$$n + m = c_{k-1} \cdot 2^k + s_{k-1} \cdot 2^{k-1} + \ldots + s_1 \cdot 2 + s_0$$

Demuestre que $c_i \leq 1$ (sin importar la base).

... por lo tanto, $|(n+m)_b| \leq \max\{|(n)_b|, |(m)_b|\} + 1$

¿cómo sumamos dos números en base b?

Ejemplo

Considere la suma de $(11)_2 = 1011$ y $(14)_2 = 1110$.

$$\begin{array}{rcl} 1+0 & = & 0\cdot 2+\mathbf{1} \\ 1+1+0 & = & 1\cdot 2+\mathbf{0} \\ 0+1+1 & = & 1\cdot 2+\mathbf{0} \\ 1+1+1 & = & 1\cdot 2+\mathbf{1} \\ 0+0+1 & = & 0\cdot 2+\mathbf{1} \end{array}$$

Por lo tanto, $(11+14)_2 = 1\,1001$.

Algoritmo de suma de números en base b

Algoritmo

input : Números n y m con $(n)_b = n_{k-1} \dots n_1 n_0$, $(m)_b = m_{k-1} \dots m_1 m_0$

output: Una secuencia $(n+m)_b = s_k s_{k-1} \dots s_1 s_0$

Function `SumaEnBaseB` (n, m)

  $c := 0$

  for $j = 0$ **to** $k-1$ **do**

    $s_j := (n_j + m_j + c)$ **mod** b

    $c := (n_j + m_j + c)$ **div** b

  $s_k := c$

  return $s_k s_{k-1} \dots s_1 s_0$

¿cuál es el **tiempo** del algoritmo en términos de k?

¿cómo multiplicamos dos números en base b?

Considere dos números en base 2 ($b = 2$):

$$\begin{aligned} n &= n_{k-1}2^{k-1} + \ldots + n_1 \cdot 2 + n_0 \\ m &= m_{k-1}2^{k-1} + \ldots + m_1 \cdot 2 + m_0 \end{aligned}$$

Multiplicando ambos números tenemos que:

$$\begin{aligned} n \cdot m &= n \cdot (m_{k-1}2^{k-1} + \ldots + m_1 \cdot 2 + m_0) \\ &= n \cdot (m_{k-1}2^{k-1}) + \ldots + n \cdot (m_1 \cdot 2) + n \cdot (m_0) \end{aligned}$$

¿cuál es el valor de $p_i := n \cdot (m_i \cdot 2^i)$?

¿cómo multiplicamos dos números en base b?

Multiplicando ambos números tenemos que:

$$n \cdot m \quad = \quad n \cdot (m_{k-1} 2^{k-1}) + \ldots + n \cdot (m_1 \cdot 2) + n \cdot (m_0)$$

¿cuánto vale $p_i := n \cdot (m_i \cdot 2^i)$?

- Si $m_i = 0$, entonces $p_i := n \cdot (m_i \cdot 2^i) = 0$.
- Si $m_i = 1$, entonces $p_i := n \cdot (m_i \cdot 2^i) = n_{k-1} 2^{i+k-1} + \ldots + n_1 \cdot 2^{i+1} + n_0 \cdot 2^i$.

$$(p_i)_2 \;=\; \begin{cases} 0 & \text{si } m_i = 0 \\ n_{k-1} \ldots n_1 n_0 \underbrace{0 \ldots 0}_{i-\text{veces}} & \text{si } m_i = 1 \end{cases}$$

Es posible calcular p_i haciendo **shift** i-veces de n.

¿cómo multiplicamos dos números en base b?

Ejemplo

Considere la multiplicación de $(14)_2 = 1110$ y $(11)_2 = 1011$.

$$
\begin{array}{rrcccr}
 & p_0 & = & 1110 \cdot (\mathbf{1} \cdot 2^0) & = & 1110 \\
 & p_1 & = & 1110 \cdot (\mathbf{1} \cdot 2^1) & = & 1110\underline{0} \\
 & p_2 & = & 1110 \cdot (\mathbf{0} \cdot 2^2) & = & 0000\underline{00} \\
+ & p_3 & = & 1110 \cdot (\mathbf{1} \cdot 2^3) & = & 1110\underline{000} \\
\hline
 & & & p_1 + p_2 + p_3 + p_4 & = & 10011010
\end{array}
$$

Por lo tanto, $(14 \cdot 11)_2 = 1001\,1010$.

Algoritmo de multiplicación de números en base b

Algoritmo

input : Números n y m con $(n)_b = n_{k-1} \dots n_1 n_0$, $(m)_b = m_{k-1} \dots m_1 m_0$
output: Una secuencia $(n \cdot m)_b = p_{2k} \dots p_1 p_0$

Function `MultiplicaciónEnBaseB` (n, m)
    for $i = 0$ **to** $k - 1$ **do**
        if $m_i > 0$ **then**
            $p_i := (n \cdot m_i)_b \underbrace{0 \dots 0}_{i-\text{veces}}$
        else
            $p_i := 0$
    $p := 0$
    for $i = 0$ **to** $k - 1$ **do**
        $p := p + p_i$
    return $(p)_b$

¿cuál es el **tiempo** del algoritmo en términos de k?

Resumen de tiempos de algoritmos

Teorema

Para números con k-dígitos en base b:

1. Suma: $\mathcal{O}(k)$.
2. Multiplicación: $\mathcal{O}(k^2)$.

¿cómo suma y multiplica el computador?