M342 Álgebra Computacional

Christian Lomp

FCUP

14 de setembro de 2011

2. Tipos de dados

2.1 Representação de número nos computador e limitações

Representação $\mathbf{q}$-ária

Seja $\mathbf{q}$ um número positivo. Qualquer número positvo $x \in \mathbb{Z}$ tem uma representação única:

$$x = (-1)^s a_0 + a_1\mathbf{q} + a_2\mathbf{q}^2 + a_3\mathbf{q}^3 + \cdots a_n\mathbf{q}^n$$

onde $a_i \in \{0, 1, \ldots, \mathbf{q} - 1\}$ e $s \in \{0, 1\}$.

Representamos um inteiro x pelo sinal s e pelo vector dos coeficientes de representação $\mathbf{q}$-ária, i.e.

$$x = (-1)^s a_n a_{n-1} \cdot a_{1_{\mathbf{q}}}$$

Exemplo

- $(342)_{10} = 1 \cdot 2^8 + 1 \cdot 2^6 + 1 \cdot 2^4 + 1 \cdot 2^2 + 1 \cdot 2^1 = (101010110)_2$
- $(342)_{10} = 1 \cdot 3^5 + 1 \cdot 3^4 + 2 \cdot 3^2 = (110200)_3$
- $(342)_{10} = 1 \cdot 16^2 + 5 \cdot 16^1 + 6 \cdot 16^0 = (156)_{16}$
- $(342)_{10} = 5 \cdot 8^2 + 2 \cdot 8^1 + 6 \cdot 8^0 = (526)_8$
- $(342)_8 = 3 \cdot 8^2 + 4 \cdot 8^1 + 2 \cdot 8^0 = (226)_{10} = (1401)_5 = (E2)_{16}$

2.

2.1 Representação de número nos computador e limitações

O tipo *unsigned int* pode representar números entre 0 e $2^{32}-1$. Seja $\mathbf{q} = 2^{32}$.

Representação 2^{32}-ária

Identificamos um tuple $x = (s, v)$ onde $s \in \{0, 1\}$
$v = \mathrm{vector}(a_0, a_1, \ldots, a_n)$ onde $0 \le a_i < 2^{32}$ com o número

$$(-1)^s a_0 + a_1 2^{32} + a_2 2^{64} + a_3 2^{96} + \cdots a_n 2^{32n}$$

2.

2.1 Representação de inteiros no computador

```
#include <vector>

class inteiros {

  int s;
  vector<int> v;

  inteiros (int sign, vector<int> coeficents)
  {
    s = sign;
    v = coeficents;
  }
}
```

Classes

Classes

1. Pode-se dizer que *Classes* são estruturas de dados com funções pre-definidos.

Classes

1. Pode-se dizer que *Classes* são estruturas de dados com funções pre-definidos.
2. Uma classe contem um conjunto de variáveis (atributos) e certas funções (métodos).

Classes

1. Pode-se dizer que *Classes* são estruturas de dados com funções pre-definidos.
2. Uma classe contem um conjunto de variáveis (atributos) e certas funções (métodos).
3. Os atriubutos e métodos podem ser privados (*private*) ou públicos (*public*).

STL vector

A classe *vector* é uma classe padrão de $C++$ e permite guardar uma lista de objectos (no caso de *vector* $<$*int*$>$ é uma lista de inteiros). Alguns funções que *vector* oferece são:

1. *push_back* : adicionar um elemento no fim da lista
2. *size* : tamanho da lista
3. *insert*: inserir elemento numa posição

```
vector<int> lista;
lista.push_back(2);
lista.push_back(3);
cout << "No.Elementos:" <<  lista.size() << endl;
cout << "O primeiro elemento: " << lista[0] << endl;
```

2.

Adição de inteiros

Sejam $x = \sum_{i=0}^{n} a_i \mathbf{q}^i$ e $y = \sum_{j=0}^{n} b_j \mathbf{q}^j$. Então

$$x + y = \sum_{i=0}^{n} (a_i + b_i)\mathbf{q}^i.$$

Como $0 \leq a_i, b_i \leq \mathbf{q} - 1$, $0 \leq a_i + b_i \leq 2\mathbf{q} - 2$.
Seja $\gamma_{-1} = 0$ e para todo $i \geq 0$ define

$$\gamma_i = \begin{cases} 0 & \text{se } a_i + b_i + \gamma_{i-1} < \mathbf{q} \\ 1 & \text{se } a_i + b_i + \gamma_{i-1} \geq \mathbf{q} \end{cases} \qquad c_i = \begin{cases} a_i + b_i + \gamma_{i-1} & \text{se } \gamma_i = 0 \\ a_i + b_i + \gamma_{i-1} - \mathbf{q} & \text{se } \gamma_i = 1 \end{cases}$$

Portanto

$$x + y = \sum_{i=0}^{n+1} c_i \mathbf{q}^i.$$

2.

Adição de inteiros

Input: inteiros $x = (s, (a_0, \ldots, a_n))$, $y = (s, (b_0, \ldots, b_n))$
Output: inteiro $z = (s, (c_0, \ldots, c_{n+1}))$ tal que $z = x + y$.

$\gamma \leftarrow 0$
for $i = 0, \ldots, n$ **do**
  $c_i \leftarrow a_i + b_i + \gamma$
  if $c_i \geq 2^{32}$ **then**
    $c_i \leftarrow c_i - 2^{32}$
    $\gamma \leftarrow 1$
  else
    $\gamma \leftarrow 0$
  end if
end for
$c_{n+1} \leftarrow \gamma$

Exemplo

$$x = (0, (2^{32} - 1, 1)) \qquad ; \quad 2^{32} - 1 + 2^{32} = 858993458$$
$$y = (0, (2^{32} - 2, 2^{32} - 1)) \qquad ; \quad 2^{32} - 2 + (2^{32} - 1) * 2^{32} = 18446744073709551614.$$

i	c_i	γ	
0	$2^{32} - 3$	$\gamma \leftarrow 1$	pois $2^{32} - 1 + 2^{32} - 2 \geq 2^{32}$
1	1	$\gamma \leftarrow 1$	pois $1 + 2^{32} - 1 + 1 \geq 2^{32}$
2	1		

Logo $x + y = (0, (2^{32} - 3, 1, 1))$ e

$$2^{32} - 3 + 2^{32} + 2^{64} = (2^{32} - 1 + 2^{32}) + (2^{64} - 2) = 18446744074568544072.$$

2.

Comparar inteiros

Igualdade

Dois inteiros $x = (s, (a_0, \ldots, a_n))$ e $y = (s, (b_0, \ldots, b_m))$ são iguais se

$$s = t \qquad n = m \qquad a_i = b_i \ \forall 0 \leq i \leq n.$$

2.

Comparar inteiros

Ordem

O inteiro $x = (s, (a_0, \ldots, a_n))$ e menor do que $y = (s, (b_0, \ldots, b_m))$ se e só se

$s < t$ ou

$(s = t) \wedge (n < m)$ ou

$(s = t) \wedge (n = m) \wedge \exists 0 \leq i \leq n : a_i < b_i \wedge \forall i < j \leq n : a_j = b_j.$

2.

Subtracção de inteiros

Input: inteiros $x = (s, (a_0, \ldots, a_n))$, $y = (s, (b_0, \ldots, b_n))$ tal que $y < x$.

Output: inteiro $z = (s, (c_0, \ldots, c_{n+1}))$ tal que $z = x - y$.

$\gamma \leftarrow 0$
for $i = 0, \ldots, n$ **do**
  $c_i \leftarrow a_i - b_i - \gamma$
  if $c_i \geq 0$ **then**
    $c_i \leftarrow c_i + 2^{32}$
    $\gamma \leftarrow 1$
  else
    $\gamma \leftarrow 0$
  end if
end for

2.

Multiplicação de inteiros

Seja $\mathbf{q} = 2^{32}$.

Multiplicação

Sejam $x = \sum_{i=0}^{n} a_i \mathbf{q}^i$ e $y = \sum_{j=0}^{m} b_j \mathbf{q}^j$. Então

$$xy = \sum_{i=0}^{n} \sum_{j=0}^{m} a_i b_j \mathbf{q}^{i+j}.$$

Como $0 \leq a_i, b_j < \mathbf{q}$, tem-se que $a_i b_j = c_{i,j} + d_{i,j}\mathbf{q}$ onde $0 \leq c_{i,j}, d_{i,j} < \mathbf{q}$.

2.

Multiplicação de inteiros

Suponha que existe uma função **mult**(*int* a,*int* b) cujo resultado é um par (c, d) de **int** tal que

$$ab = c + 2^{32}d$$

Input: inteiros $x = (s, (a_0, \ldots, a_n))$, $y = (t, (b_0, \ldots, b_m))$.
Output: inteiro $z = (u, (c_0, \ldots, c_{nm}))$ tal que $z = xy$.

$z \leftarrow ((s + t)\mathrm{mod}2, (0))$
for $i = 0, \ldots, n$ **do**
  for $j = 0, \ldots, m$ **do**
    $(c, d) \leftarrow \mathbf{mult}(a_i, b_j)$
    $z \leftarrow z + c2^{i+j}$
    $z \leftarrow z + d2^{i+j+1}$
  end for
end for

2.

Divisão de inteiros