Целочисленная арифметика многократной точности

Эдвард Гогин

14 декабря, 2023, Москва, Россия

Российский Университет Дружбы Народов

Цели и задачи

Цель лабораторной работы

Ознакомление с алгоритмами целочисленной арифметики многократной точности, а также их последующая программная реализация.

Выполнение лабораторной работы

Длинная арифметика

Высокоточная (длинная) арифметика — это операции (базовые арифметические действия, элементарные математические функции и пр.) над числами большой разрядности (многоразрядными числами), т.е. числами, разрядность которых превышает длину машинного слова универсальных процессоров общего назначения (более 128 бит).

Сложение неотрицательных целых чисел

- Вход. Два неотрицательных числа $u = u_1 u_2 \dots u_n$ и $v = v_1 v_2 \dots v_n$; разрядность чисел n; основание системы счисления b.
- Выход. Сумма $w = w_0 w_1 \dots w_n$, где w_0 - цифра переноса, всегда равная 0 либо 1.

1. Присвоить $j = n, k = 0$ (j *идет по разрядам,* k *следит за переносом*).
2. Присвоить $w_j = (u_j + v_j + k) \pmod b$, где $k = \left[\frac{u_j + v_j + k}{b}\right]$.
3. Присвоить $j = j - 1$. Если $j > 0$, то возвращаемся на шаг 2; если $j = 0$, то присвоить $w_0 = k$ и результат: w.

Вычитание неотрицательных целых чисел

- Вход. Два неотрицательных числа $u = u_1 u_2 \dots u_n$ и $v = v_1 v_2 \dots v_n$, $u > v$; разрядность чисел n; основание системы счисления b.
- Выход. Разность $w = w_0 w_1 \dots w_n = u - v$.

1. Присвоить $j = n$, $k = 0$ (k – заём из старшего разряда).
2. Присвоить $w_j = (u_j - v_j + k) \pmod b$; $k = \left[\frac{u_j - v_j + k}{b}\right]$.
3. Присвоить $j = j - 1$. Если $j > 0$, то возвращаемся на шаг 2; если $j = 0$, то результат: w.

Умножение неотрицательных целых чисел столбиком

- Вход. Числа $u = u_1 u_2 \dots u_n$, $v = v_1 v_2 \dots v_m$; основание системы счисления b.
- Выход. Произведение $w = uv = w_1 w_2 \dots w_{m+n}$.

1. Выполнить присвоения:
   $w_{m+1} = 0, w_{m+2} = 0, \dots, w_{m+n} = 0, j = m$ (*j перемещается по номерам разрядов числа v от младших к старшим*).
2. Если $v_j = 0$, то присвоить $w_j = 0$ и перейти на шаг 6.

Умножение неотрицательных целых чисел столбиком

3. Присвоить $i = n, k = 0$ (*значение* i *идет по номерам разрядов числа* u, k *отвечает за перенос*).
4. Присвоить $t = u_i \cdot v_j + w_{i+j} + k, w_{i+j} = t \pmod b, k = \left[\frac{t}{b}\right]$.
5. Присвоить $i = i - 1$. Если $i > 0$, то возвращаемся на шаг 4, иначе присвоить $w_j = k$.
6. Присвоить $j = j - 1$. Если $j > 0$, то вернуться на шаг 2. Если $j = 0$, то результат: w.

Быстрый столбик

- Вход. Числа $u = u_1 u_2 \dots u_n$, $v = v_1 v_2 \dots v_m$; основание системы счисления b.
- Выход. Произведение $w = uv = w_1 w_2 \dots w_{m+n}$.

1. Присвоить $t = 0$.
2. Для s от 0 до $m + n - 1$ с шагом 1 выполнить шаги 3 и 4.
3. Для i от 0 до s с шагом 1 выполнить присвоение $t = t + u_{n-i} \cdot v_{m-s+i}$.
4. Присвоить $w_{m+n-s} = t \pmod b$, $t = \left[\frac{t}{b}\right]$. Результат: w.

Деление многоразрядных целых чисел

- Вход. Числа $u = u_n \dots u_1 u_0$, $v = v_t \dots v_1 v_0, n \geq t \geq 1, v_t \neq 0$.
- Выход. Частное $q = q_{n-t} \dots q_0$, остаток $r = r_t \dots r_0$.

1. Для j от 0 до $n - t$ присвоить $q_j = 0$.
2. Пока $u \geq vb^{n-t}$, выполнять: $q_{n-t} = q_{n-t} + 1, u = u - vb^{n-t}$.
3. Для $i = n, n-1, \dots, t+1$ выполнять пункты 3.1 – 3.4: 3.1. если $u_i \geq v_t$, то присвоить $q_{i-t-1} = b - 1$, иначе присвоить $q_{i-t-1} = \frac{u_i b + u_{i-1}}{v_t}$. 3.2. пока $q_{i-t-1}(v_t b + v_{t-1}) > u_i b^2 + u_{i-1} b + u_{i-2}$ выполнять $q_{i-t-1} = q_{i-t-1} - 1$. 3.3. присвоить $u = u - q_{i-t-1} b^{i-t-1} v$. 3.4. если $u < 0$, то присвоить $u = u + vb^{i-t-1}, q_{i-t-1} = q_{i-t-1} - 1$.
4. $r = u$. Результат: q и r.

Пример работы алгоритма

```
    else:
        q[i-t-1] = math.floor((int(u[i])*b + int(u[i-1]))/int(v[t]))

    while (int(q[i-t-1])*(int(v[t])*b + int(v[t-1])) > int(u[i])*(b**2) + int(u[i-1])*b + int(u[i-2])):
        q[i-t-1] = q[i-t-1] - 1
    u = (int(u) - q[i-t-1]*b**(i-t-1)*int(v))
    if u < 0:
        u = int(u) + int(v) *(b**(i-t-1))
        q[i-t-1] = q[i-t-1] - 1
r = u
print(q, r)

```

```
[6, 9, 1, 3, 4]
[4, 4, 4, 4, 4]
[0, 0, 0, 0, 0, 0, 0.3999999999999986, 4, 0, 0]
[8, 3, 1, 4, 0, 2, 0, 5, 0, 0, 0]
[0, 2, 9] -39899091
```

Figure 1: Работа алгоритма

Выводы

Результаты выполнения лабораторной работы

Изучили алгоритмы целочисленной арифметики.