Отчёт по лабораторной работе №8. Целочисленная арифметика многократной точности

Дисциплина: Математические основы защиты информации и информационной безопасности

Студент: Майорова О.А., 1032212322
Группа: НФИмд-02-21
Преподаватель: д.ф.-м.н., Кулябов Д. С.

Москва, 2021

Цель и задачи

Цель: Ознакомиться с целочисленной арифметикой многократной точности.

Задача: Программно реализовать алгоритмы: сложения неотрицательных целых чисел, вычитания неотрицательных целых чисел, умножения неотрицательных целых чисел, быстрый столбик и деления многоразрядных целых чисел.

Сложение неотрицательных целых чисел

Алгоритм 1 (сложение неотрицательных целых чисел).

Вход. Два неотрицательных числа $u = u_1u_2 \dots u_n$ и $v = v_1v_2 \dots v_n$; разрядность чисел n; основание системы счисления b.

Выход. Сумма $w = w_0w_1 \dots w_n$, 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 w_j – наименьший неотрицательный вычет в данном классе вычетов; $k = \left[\frac{u_j+v_j+k}{b}\right]$.
3. Присвоить $j := j - 1$. Если $j > 0$, то возвращаемся на шаг 2; если $j = 0$, то присвоить $w_0 := k$ и результат: w.


```python
def alg1(u, v, n, b):
    u = [int(x) for x in str(u)]
    v = [int(x) for x in str(v)]
    w = []
    j = n
    k = 0

    while j > 0:
        w.append((u[j-1] + v[j-1] + k) % b)
        k = (u[j-1] + v[j-1] + k) // b
        j -= 1

    w.append(k)
    w.reverse()

    return int(''.join(str(x) for x in w))
```

Результат:

```
1  alg1(4773, 4237, 4, 10), 4773+4237
```

```
(9010, 9010)
```

```
1  alg1(4321, 1234, 4, 10), 4321+1234
```

```
(5555, 5555)
```

Вычитание неотрицательных целых чисел

Алгоритм 2 (вычитание неотрицательных целых чисел).

Вход. Два неотрицательных числа $u = u_1u_2 \dots u_n$ и $v = v_1v_2 \dots v_n$, $u > v$; разрядность чисел n; основание системы счисления b.

Выход. Разность $w = w_1w_2 \dots w_n = u - v$.

1. Присвоить $j := n$, $k := 0$ (k – заем из старшего разряда).
2. Присвоить $w_j = (u_j - v_j + k) \ (mod\ b)$, где w_j – наименьший неотрицательный вычет в данном классе вычетов; $k = \left[\frac{u_j - v_j + k}{b}\right]$.
3. Присвоить $j := j - 1$. Если $j > 0$, то возвращаемся на шаг 2; если $j = 0$, то результат: w.

```python
def alg2(u, v, n, b):
    u = [int(x) for x in str(u)]
    v = [int(x) for x in str(v)]
    w = []
    j = n
    k = 0

    while j > 0:
        w.append((u[j-1] - v[j-1] + k) % b)
        k = (u[j-1] - v[j-1] + k) // b
        j -= 1

    w.reverse()

    return int(''.join(str(x) for x in w))
```

Результат:

```
1  alg2(4773, 4237, 4, 10), 4773-4237
```

```
(536, 536)
```

```
1  alg2(4321, 1234, 4, 10), 4321-1234
```

```
(3087, 3087)
```

Умножение неотрицательных целых чисел

Алгоритм 3 (умножение неотрицательных целых чисел столбиком).

Вход. Числа $u = u_1u_2 \dots u_n$, $v = v_1v_2 \dots v_m$; основание системы счисления b.

Выход. Произведение $w = uv = w_1w_2 \dots w_{m+n}$.

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.
3. Присвоить $i := n, k := 0$ (Значение i идет по номерам разрядов числа u, k отвечает за перенос).
4. Присвоить $t := u_i \cdot v_j + w_{i+j} + k$, $w_{i+j} := t \,(mod\, b)$, $k := \frac{t}{b}$, где w_{i+j} – наименьший неотрицательный вычет в данном классе вычетов.
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.


```python
def alg3(u, v, b):
    u = [int(x) for x in str(u)]
    v = [int(x) for x in str(v)]
    w = [0] * len(u + v)
    j = len(v)
    k = 0

    while j > 0:
        if v[j-1] == 0:
            w[j-1] = 0
        else:
            i = len(u)
            k = 0
            while i > 0:
                t = u[i-1]*v[j-1] + w[i+j-1] + k
                w[i+j-1] = t % b
                k = t // b
                i -= 1
            w[j-1] = k
        j -= 1

    return int(''.join(str(x) for x in w))
```

Результат:

```
1  alg3(4773, 4237, 10), 4773*4237
```

```
(20223201, 20223201)
```

```
1  alg3(4321, 1234, 10), 4321*1234
```

```
(5332114, 5332114)
```

Быстрый столбик

Алгоритм 4 (быстрый столбик).

Вход. Числа $u = u_1u_2 \dots u_n$, $v = v_1v_2 \dots v_m$; основание системы счисления b.

Выход. Произведение $w = uv = w_1w_2 \dots w_{m+n}$.

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 $w_{m+n-s} := t \ (mod\ b)$, $t := \frac{t}{b}$, где w_{m+n-s} – наименьший неотрицательный вычет по модулю b. Результат: w.

```python
def alg4(u, v, b):
    u = [int(x) for x in str(u)]
    v = [int(x) for x in str(v)]
    w = [0] * len(u + v)
    t = 0
    n = len(u)
    m = len(v)

    j = -n + 1
    for s in range(m + n):
        if s >= (m + n) // 2:
            r1 = j
            r2 = s+1-j
        else:
            r1 = 0
            r2 = s+1

        for i in range(r1, r2):
            t += u[n-i-1] * v[m-s+i-1]

        j += 1
        w[m+n-s-1] = t % b
        t = t // b

    return int(''.join(str(x) for x in w))
```

Результат:

```
1  alg4(4773, 4237, 10), 4773*4237
```

```
(20223201, 20223201)
```

```
1  alg4(4321, 1234, 10), 4321*1234
```

```
(5332114, 5332114)
```

Деление многоразрядных целых чисел

Алгоритм 5 (деление многоразрядных целых чисел).

Вход. Числа $u = u_n \dots u_1 u_0$, $v = v_t \dots v_1 v_0$, $n \geq t \geq 1$, $v_t \neq 0$, разрядность чисел соответственно n и t.

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.


```python
def alg5(u, v, b):
    n = len(str(u))
    t = len(str(v))
    q = [0] * (n - t + 1)

    while u >= v*b**(n-t):
        q[n-t] += 1
        u -= v*b**(n-t)

    ul = [int(x) for x in str(u)]
    vl = [int(x) for x in str(v)]

    i = n
    while i > t:
        if ul[i] >= vl[t]:
            q[i-t-1] = b-1
        else:
            q[i-t-1] = (ul[i]*b + ul[i-1])/vl[t]

        while q[i-t-1]*(vl[t]*b + vl[t-1]) > ul[i] * b**2 + ul[i-1]*b + ul[i-2]:
            q[i-t-1] -= 1

        u = u - q[i-t-1] * b**(i-t-1) * v
        ul = [int(x) for x in str(u)]

        if u < 0:
            u = u + v * b**(i-t-1)
            ul = [int(x) for x in str(u)]

            q[i-t-1] -= 1

        i -= 1

    return int(''.join(str(x) for x in q)), u
```

Результат:

```
1  alg5(4773, 4237, 10), 4773//4237, 4773%4237
```

```
((1, 536), 1, 536)
```

```
1  alg5(4321, 1234, 10), 4321//1234, 4321%1234
```

```
((3, 619), 3, 619)
```

Заключение

Таким образом, была достигнута цель, поставленная в начале лабораторной работы.

- Было осуществлено знакомство с целочисленной арифметикой многократной точности.
- Также была получена реализация на языке Python алгоритмов сложения неотрицательных целых чисел, вычитания неотрицательных целых чисел, умножения неотрицательных целых чисел, быстрый столбик и деления многоразрядных целых чисел.

Спасибо за внимание