

Математические основы защиты информации и информационной безопасности. Лабораторная работа № 8 на тему “Целочисленная арифметика многократной точности”

Лубышева Ярослава Михайловна

RUDN University, Moscow, Russian Federation

Содержание

- Цели и задачи
- Выполнение
- Результаты
- Список литературы

Цели и задачи

Выполнить задание к лабораторной работе № 8:

1. Ознакомиться с алгоритмами: сложение неотрицательных целых чисел, вычитание неотрицательных целых чисел, умножение неотрицательных целых чисел столбиком, умножение неотрицательных целых чисел столбиком (быстрый столбик), деление многоразрядных целых чисел
2. Реализовать алгоритмы программно

Выполнение

```
# 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 равна 0 или 1.
def plus(u, v, n, b):
    j = n-1 # идет по разрядам (-1 так как индексирование списков начинается с 0)
    k = 0 # следит за переносом
    w = [None] * n

    while j >= 0:
        w[j] = (u[j]+v[j]+k) % b # наименьший неотриц. вычет в данном классе вычетов
        k = (u[j]+v[j]+k) // b
        j -= 1

    w.insert(0, k)
    return w
```

```
# 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: разрядность  $w=w_1w_2\dots w_n = u - v$ .
def minus(u, v, n, b):
    if u <= v:
        return "Введите другие числа: первое число должно быть больше второго"

    j = n-1 # так как индексирование списков начинается с 0
    k = 0 #заем из старшего разряда
    w = [None] * n

    while j >= 0:
        w[j] = (u[j]-v[j]+k) % b # наименьший неотриц. вычет в данном классе вычетов
        k = (u[j]-v[j]+k) // b
        j -= 1

    return w
```



```
# Умножение неотрицательных целых чисел столбиком
# Вход: числа u=u_1u_2...u_n и v=v_1v_2...v_m; основание системы счисления b;
# Выход: произведение w = u*v = w_1w_2...w_(m+n).
def long_multiplication(u, v, b):
    n = len(u)
    m = len(v)
    w = [0] * (m+n)
    # j перемещается по номерам разрядов числа v от младших к старшим
    # условие шага 6
    for j in range(m-1, -1, -1):
        # шаг 2
        if v[j] != 0:
            # шаг 3
            k = 0 # отвечает за перенос
            # условие шага 5
            for i in range(n-1, -1, -1):
                # шаг 4
                t = u[i] * v[j] + w[i+j+1] + k
                w[i+j+1] = t % b # наименьший неотриц. вычет в данном классе вычетов
                k = t // b
            # если i==0
            w[j] = k
    return w
```

```
# Умножение неотрицательных целых чисел столбиком (быстрый столбик)
# Вход: числа  $u = u_1u_2...u_n$  и  $v = v_1v_2...v_m$ ; основание системы счисления  $b$ ;
# Выход: произведение  $w = u*v = w_1w_2...w_{(m+n)}$ .
def quick_long_multiplication(u, v, b):
    n = len(u)
    m = len(v)
    w = [0] * (m+n)
    # шаг 1
    t = 0
    # шаг 2
    for s in range(0, m+n):
        # шаг 3
        for i in range(0, s+1):
            if 0<=n-i-1<n and 0<=m-s+i-1<m:
                t += u[n-i-1] * v[m-s+i-1]
        # шаг 4
        w[m+n-s-1] = t % b #наименьший неотрицательный вычет по модулю b
        t //= b
    return w
```

```
# Преобразование числа-списка в int
def list_to_int(lst):
    return int(''.join(map(str, lst)))

# преобразование int в число-строку
# len - количество цифр в числе-строке до преобразования в int
def int_to_list(num, num_size):
    num_list = list(map(int, str(num)))
    # добавляем недостающие нули перед значением
    num_list = [0]*(num_size - len(num_list)) + num_list
    return num_list
```

```
# Деление многоразрядных целых чисел
# Вход: числа u=u_n...u_1u_0, v=v_t...v_1v_0, n>=t>=1, v_t!=0,
# где n и t -разрядность чисел; основание системы счисления b;
# Выход: частное q=q_(n-t)...q_0, остаток r=r_t...r_0.
def dividing_multi_digit_integers(u, v, b):
    n = len(u) - 1
    t = len(v) - 1
    q = [0] * (n-t+1)
    # war 2
    u, v = list_to_int(u), list_to_int(v) # преобразуем в int
    while u >= v*b**(n-t):
        q[n-t] += 1
        u -= v*b**(n-t)
    # war 3
    for i in range(n, t, -1):
        u, v = int_to_list(u, n+1), int_to_list(v, t+1) # преобразуем в list
        # war 3.1
        if u[n-i]>v[0]:
            q[i-t-1] = b - 1
        else:
            q[i-t-1] = (u[n-i]*b + u[n-i+1]) // v[0]
        # war 3.2
        while q[i-t-1]*(v[0]*b + v[1]) > u[n-i]*b**2 + u[n-i+1]*b + u[n-i+2]:
            q[i-t-1] -= 1
        # war 3.3
        u, v = list_to_int(u), list_to_int(v) # преобразуем в int
        u -= q[i-t-1] * b**(i-t-1) * v
        # war 3.4
        if u<0:
            u += v*b**(i-t-1)
            q[i-t-1] -= 1
    # war 4
    r = u
    return q, r
```

```
u = [3, 0, 6]
v = [1, 0, 4]
n = 3
b = 10

print(f"{plus(u, v, n, b)}")
print(f"{minus(u, v, n, b)}")
print(f"{long_multiplication([1, 1], [1, 1], b)}")
print(f"{quick_long_multiplication([1, 1], [1, 1], b)}")
print(f"{dividing_multi_digit_integers([1, 2, 7, 8], [4, 1, 9], b)}")

[0, 4, 1, 0]
[2, 0, 2]
[0, 1, 2, 1]
[0, 1, 2, 1]
([3, 0], 21)
```

Результаты

Выполнено задание к лабораторной работе № 8

Список литературы

1. Методические материалы курса