

Отчёт по лабораторной работе №8

**Дисциплина: Математические основы защиты информации и
информационной безопасности**

Полиенко Анастасия Николаевна, НПМмд-02-23

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	6
4	Выполнение лабораторной работы	7
5	Выводы	11
	Список литературы	12

Список иллюстраций

4.1	Алгоритм 1	7
4.2	Алгоритм 2	8
4.3	Алгоритм 4	9
4.4	Алгоритм 5	10

1 Цель работы

Изучить алгоритмы целочисленной арифметики многократной точности.

2 Задание

Реализовать пять алгоритмов целочисленной арифметики:

1. Сложение неотрицательных целых чисел
2. Вычитание неотрицательных целых чисел
3. Умножение неотрицательных целых чисел столбиком
4. Быстрый столбик
5. Деление многоразрядных целых чисел

3 Теоретическое введение

Будем считать, что число записано в b -ичной системе счисления, b - натуральное число, $b \geq 2$. Натуральное n -разрядное число будем записывать в виде

$$u = u_1 u_2 \dots u_n$$

При работе с большими целыми знак такого удобно хранить в отдельной переменной. Например, при умножении двух чисел, знак произведения вычисляется отдельно.

Более подробно см. в [1–6].

4 Выполнение лабораторной работы

1. Реализуем алгоритм сложения неотрицательных целых чисел (рис. 4.1).

```
a = 0;  
b = pi/2;  
n = 100;  
dx = (b-a)/n;  
  
function y = f(x)  
    y = exp(x.^2) .* cos(x);  
end  
  
m = [a + dx/2:dx:b - dx/2];  
M = f(m)  
approx = dx * sum (M)
```

Рис. 4.1: Алгоритм 1

2. Реализуем алгоритм вычитания неотрицательных целых чисел (рис. 4.2).

```

Ввод [4]: b = 10
n = 5
u = [1, 2, 3, 4, 5]
v = [6, 7, 8, 9, 0]
w = [0]*(n + 1)

k = 0
for j in range(n-1, -1, -1):
    w[j + 1] = (u[j] + v[j] + k) % b
    k = (u[j] + v[j] + k) // b

w[0] = k
print(w)

[0, 8, 0, 2, 3, 5]

```

Рис. 4.2: Алгоритм 2

3. Реализуем алгоритм умножения неотрицательных целых чисел столбиком (рис. ??).

```

b = 10
n = 5
u = [1, 2, 3, 4, 5]
m = 3
v = [1, 2, 3]
w = [0]*(m + n)

for j in range(m-1, -1, -1):
    if v[j] == 0:
        w[j] = 0
    else:
        k = 0
        for i in range(n-1, -1, -1):
            t = u[i] * v[j] + w[i + j + 1] + k
            w[i + j + 1] = t % b
            k = t // b
        w[j] = k

print(w)

[0, 1, 5, 1, 8, 4, 3, 5]

```

4. Реализуем алгоритм быстрого столбика (рис. 4.3).


```

b = 10
n = 4
u = [1, 2, 3, 4, 5]
m = 2
v = [1, 2, 3]
w = [0]*(len(u) + len(v))

t = 0
for s in range(m + n + 2):
    for i in range(s + 1):
        if (n - i < 0) or (m - s + i < 0):
            t = t
        else:
            t = t + u[n - i] * v[m - s + i]
    w[m + n - s + 1] = t % b
    t = t // b

print(w)

```

[0, 1, 5, 1, 8, 4, 3, 5]

Рис. 4.3: Алгоритм 4

5. Реализуем алгоритм деления многоразрядных целых чисел (рис. 4.4).

```

b = 10
u = 12345
v = 123
n = 5
t = 3

q = [0] * (n - t + 1)
r = [0] * (t + 1)

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

for i in range(n, t+2, -1):
    if u[i] >= v[t]:
        q[i-t-1] = b - 1
    else:
        q[i-t-1] = (u[i] * b + u[i - 1]) // v[t]
        while q[i-t-1] * (v[t] * b + v[t-1]) > u[i] * b^2 + u[i-1] * b + u[i-2]:
            q[i-t-1] -= 1
        u -= q[i-t-1] * (b ** (i-t-1)) * v
        if u < 0:
            u += v * (b ** (i - t - 1))
            q[i-t-1] -= 1
r = u
print("q =", q[::-1])
print("r =", r)

```

Рис. 4.4: Алгоритм 5

5 Выводы

Изучила алгоритмы целочисленной арифметики многократной точности.

Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016.
URL: <https://www.gnu.org/software/bash/manual/>.
2. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.