# Отчет по лабораторной работе №8

**Целочисленная арифметика многократной точности**

Арам Грачьяевич Саргсян

# Содержание

**1 Цель работы** **5**

**2 Выполнение лабораторной работы** **6**

**3 Выводы** **12**

# Список иллюстраций

# Список таблиц

# 1 Цель работы

Изучить алгоритмы вычисления суммы, разницы, умножения, деления больших целых чисел.

# 2 Выполнение лабораторной работы

1. Я реализовал все вышеперечисленные алгоритмы на языке julia.

```
function list_to_number(digits)
    number = 0
    for digit in digits
        number = number * 10 + digit
    end
    return number
end

function add(u, v, b)
    n = length(u)
    m = length(v)
    k = 0
    w = Int[]

    for i in 1:max(n, m)
        ui = i <= n ? u[end - i + 1] : 0
        vi = i <= m ? v[end - i + 1] : 0
        sum = ui + vi + k
        push!(w, sum % b)
        k = div(sum, b)
    end
```

```
    if k > 0
        push!(w, k)
    end

    return list_to_number(reverse(w))
end

function subtract(u, v, b)
    n = length(u)
    m = length(v)
    k = 0
    w = Int[]

    for i in 1:n
        ui = u[end - i + 1]
        vi = i <= m ? v[end - i + 1] : 0
        diff = ui - vi + k
        if diff < 0
            diff += b
            k = -1
        else
            k = 0
        end
        push!(w, diff)
    end

    while length(w) > 1 && w[end] == 0
        pop!(w)
    end
```

```
    return list_to_number(reverse(w))
end

function multiply(u, v, b)
    n = length(u)
    m = length(v)
    w = zeros(Int, n + m)

    for i in 1:n
        carry = 0
        for j in 1:m
            product = u[n - i + 1] * v[m - j + 1] + w[i + j - 1] + carry
            w[i + j - 1] = product % b
            carry = div(product, b)
        end
        w[i + m] += carry
    end

    while length(w) > 1 && w[end] == 0
        pop!(w)
    end

    return list_to_number(reverse(w))
end

function fast_multiply(u, v, b)
    n = length(u)
    m = length(v)
    w = zeros(Int, n + m)
```

```
    for i in 1:n
        carry = 0
        for j in 1:m
            t = w[i + j - 1] + u[n - i + 1] * v[m - j + 1] + carry
            w[i + j - 1] = t % b
            carry = div(t, b)
        end
        w[i + m] += carry
    end

    while length(w) > 1 && w[end] == 0
        pop!(w)
    end

    return list_to_number(reverse(w))
end

function divide(u, v, b)
    n = length(u)
    m = length(v)
    q = zeros(Int, n - m + 1)
    r = deepcopy(u)

    for i in 0:(n - m)
        t = n - i
        q[i + 1] = div(r[t], v[m])
        while q[i + 1] * v[m] > (r[t] * b + r[t - 1])
            q[i + 1] -= 1
        end
```

```
        for j in 1:m
            r[t - j + 1] -= q[i + 1] * v[m - j + 1]
            if r[t - j + 1] < 0
                r[t - j + 1] += b
                r[t - j] -= 1
            end
        end
    end

    while length(r) > 1 && r[end] == 0
        pop!(r)
    end

    return list_to_number(q), list_to_number(r)
end

u = [1, 2, 3, 4, 5, 6, 7, 8, 9]
v = [1, 2, 3, 4, 5, 6, 7, 8, 9]
b = 10

println("Сложение: ", add(u, v, b))
println("Вычитание: ", subtract(u, v, b))
println("Умножение: ", multiply(u, v, b))
println("Быстрое умножение: ", fast_multiply(u, v, b))
q, r = divide(u, v, b)
println("Деление: q = ", q, ", r = ", r)
```

```
Сложение: 246913578
Вычитание: 0
Умножение: 15241578750190521
Быстрое умножение: 15241578750190521
```

```
Деление: q = 1, r = 0
```

# 3 Выводы

Я реализовал метод Полларда для Дискретного логарифмирования