

ЛАБОРАТОРНАЯ РАБОТА №8

**Математические основы защиты информации и
информационной безопасности. Целочисленная
арифметика многократной точности**

Кузнецов Юрий Владимирович

Введение

В данной работе рассмотрим алгоритмы для выполнения арифметических операций с большими целыми числами.

Основное содержание

Алгоритм:

- Алгоритм 1(сложение неотрицательных целых чисел).
- Алгоритм 2(вычитание неотрицательных целых чисел).
- Алгоритм 3 (умножение неотрицательных целых чисел столбиком).
- Алгоритм 4(быстрый столбик).

Кодовая реализация

Алгоритм 1(сложение неотрицательных целых чисел).

```
function add_large_numbers(a, b, base=10)
    carry = 0
    a_digits = parse.(Int, collect(string(a)))
    b_digits = parse.(Int, collect(string(b)))
    len_a = length(a_digits)
    len_b = length(b_digits)
    max_len = max(len_a, len_b)
    result = zeros(Int64, max_len + 1)

    if len_a < len_b
        a_digits = vcat(zeros(Int64, len_b - len_a), a_digits)
    elseif len_b < len_a
        b_digits = vcat(zeros(Int64, len_a - len_b), b_digits)
    end

    for i in reverse(1:max_len)
        temp = (a_digits[i] + b_digits[i] + carry) % base
        result[i+1] = temp
        carry = (a_digits[i] + b_digits[i] + carry) ÷ base
    end
end
```

```

end

result[1] = carry
return parse(Int, join(string.(result)))
end

println(add_large_numbers(87452, 1238))

```

Алгоритм 2(вычитание неотрицательных целых чисел).

```

function subtract_large_numbers(a, b, base=10)
    if a < b
        return "$a is smaller than $b"
    end

    a_digits = parse.(Int, collect(string(a)))
    b_digits = parse.(Int, collect(string(b)))
    len_a = length(a_digits)
    len_b = length(b_digits)
    max_len = max(len_a, len_b)
    result = zeros(Int64, max_len)

    if len_b < len_a
        b_digits = vcat(zeros(Int64, len_a - len_b), b_digits)
    end

    for i in reverse(1:max_len)

```

```

    if a_digits[i] < b_digits[i]
        a_digits[i-1] -= 1
        a_digits[i] += base
    end
    result[i] = a_digits[i] - b_digits[i]
end

return parse(Int, join(string.(result)))
end

println(subtract_large_numbers(87452, 1238))

```

Алгоритм 3 (умножение неотрицательных целых чисел столбиком).

```

function multiply_large_numbers(a, b, base=10)
    a_digits = parse.(Int, collect(string(a)))
    b_digits = parse.(Int, collect(string(b)))
    len_a = length(a_digits)
    len_b = length(b_digits)
    result = zeros{Int64, len_a + len_b}

    for j in reverse(1:len_b)
        carry = 0
        for i in reverse(1:len_a)
            temp = a_digits[i] * b_digits[j] + result[i+j] + carry
            result[i+j] = temp % base
            carry = temp ÷ base
        end
    end
end

```

```

        end
        result[j] = carry
    end

    return parse(Int, join(string.(result)))
end

println(multiply_large_numbers(87452, 1238))

```

Алгоритм 4(быстрый столбик).

```

function multiply_fast(a, b, base=10)
    a_digits = parse.(Int, collect(string(a)))
    b_digits = parse.(Int, collect(string(b)))
    len_a = length(a_digits)
    len_b = length(b_digits)
    result = zeros{Int64, len_a + len_b}
    temp = 0

    for s in 0:len_a+len_b-1
        for i in 0:s
            if len_a-i <= 0 || len_b-s+i <= 0
                continue
            end
            temp += a_digits[len_a-i] * b_digits[len_b-s+i]
        end
        result[len_a+len_b-s] = temp % base
        temp = temp ÷ base
    end
end

```

```
    return parse(Int, join(string.(result)))  
end
```

```
println(multiply_fast(87452, 1238))
```


Заключение

В данной лабораторной работе были реализованы алгоритмы для выполнения арифметических операций с большими целыми числами.