Лабораторная работа №8

Дискретное логарифмирование в конечном поле

Кузнецов Юрий Владимирович

Содержание I

Содержание II

Section 1

Введение

Введение

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

Алгоритм

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

Section 2

Алгоритм 1(сложение неотрицательных целых чисел).

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)
```

Алгоритм 1(сложение неотрицательных целых чисел).

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

Алгоритм 1(сложение неотрицательных целых чисел).

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

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

println(add_large_numbers(87452, 1238))
```

Алгоритм 1(сложение неотрицательных целых чисел).

```
x1, a1, b1 = update_xab(x1, a1, b1, prime_m
trace1[:, iteration] = [x1, a1, b1]


x2, a2, b2 = update_xab(x2, a2, b2, prime_m
x2, a2, b2 = update_xab(x2, a2, b2, prime_m
trace2[:, iteration] = [x2, a2, b2]

if x1 == x2
  display(trace1[:, 1:iteration])
  display(trace2[:, 1:iteration])
```

Section 3

Алгоритм 2(вычитание неотрицательных целых чисел).

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
```

Алгоритм 2(вычитание неотрицательных целых чисел).

```
    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))
```

Section 4

Алгоритм 3 (умножение неотрицательных целых чисел столбиком).

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)
```

Алгоритм 3 (умножение неотрицательных целых чисел столбиком).

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

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

Section 5

Алгоритм 4(быстрый столбик).

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
```

Алгоритм 4(быстрый столбик).

```
    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[
        end
        result[len_a+len_b-s] = temp % base
        temp = temp ÷ base
    end

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

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

Section 6

Заключение

Заключение

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