Отчёт по лабораторной работе 6

Супонина Анастасия Павловна

Содержание

Список иллюстраций

Список таблиц

Элементы списка иллюстраций не найдены.

Цель работы

Изучить $ p $-метод Полларда и научиться его программно реализовывать.

Задание

Программно реализовать на языке Julia $ p $-метод Полларда

1. Реализовать алгоритм программно.

2. Разложить на множители данное преподавателем число.

Теоретическое введение

Задача разложения составного числа на множители

Формулируется следующим образом: для данного положительного целого числа $ n $ найти его каноническое разложение:

$$n = p_1^{a_1} p_2^{a_2} \dots p_s^{a_s},$$

где $ p_i $ — попарно различные простые числа, $ a_i $.

На практике не обязательно находить каноническое разложение числа $ n .Достаточнонайтиегоразложениенадванетривиальныхсомножителя:$ n = p q, p q < n. $$
Далее будем понимать задачу разложения именно в этом смысле.

$ p $-Метод Полларда

Пусть $ n $ — нечетное составное число, $ S = {0, 1, …, n - 1} $, и $ f : S S $ — случайное отображение, обладающее сжимающими свойствами, например:

$$f(x) = x^2 + 1 \pmod n.$$

Основная идея метода состоит в следующем:

1. Выбираем случайный элемент $ x_0 S $ и строим последовательность $ x_0, x_1, x_2, ,определяемуюрекуррентнымсоотношением:$ x_{i+1} = f(x_i), $$ где $ i $, до тех пор, пока не найдем такие числа $ i, j $, что $ i < j $ и $ x_i = x_j $.

2. Поскольку множество $ S $ конечно, такие индексы $ i, j $ существуют (последовательность “зацикливается”). Последовательность $ {x_i} $ будет состоять из “хвоста” $ x_0, x_1, …, x_{l-1} $ длины $ O() $ и цикла $ x_j, x_{j+1}, …, x_{j-1} $ той же длины.

Алгоритм, реализующий $ p $-метод Полларда:

Вход:
Число $ n $, начальное значение $ c $, функция $ f $, обладающая сжимающими свойствами.

Выход:
Нетривиальный делитель числа $ n $.

1. Положить $ a c $, $ b c $.

2. Вычислить:

$$a \leftarrow f(a) \pmod n, \quad b \leftarrow f\big(f(b)\big) \pmod n.$$

3. Найти:

$$d \leftarrow \gcd(a - b, n).$$

4. Если $ 1 < d < n $, то положить $ p d $ и вернуть $ p $.
   Если $ d = n $, результат: “Делитель не найден”; при $ d = 1 $ вернуться на шаг 2.

Пример:

Найти $ p $-методом Полларда нетривиальный делитель числа $ n = 1359331 $. Положим $ c = 1 $ и $ f(x) = x^2 + 5 $. Работа алгоритма иллюстрируется следующей таблицей:

$ i $	$ a $	$ b $	$ d = (a - b, n) $
1	1	1	1
2	6	41	1
3	41	123939	1
4	1686	391594	1
5	123939	438157	1
6	435426	582738	1
7	391594	1144026	1
8	1090062	885749	1181

Результат:
1181 является нетривиальным делителем числа $ 1359331 $.

Выполнение лабораторной работы

Для использования данного алгоритма необходимо находить НОД, для этого я взяла уже сделанную программу из прошлой лабораторной работы и записала её в виде отдельной функции. А именно, для нахождения НОД я использовала “Расширенный алгоритм Евклида”

```
function euclid(n, number1)
    r_array = [number1, n]
    x_array = Float64[1, 0]
    y_array = Float64[0, 1]
    i = 1
    while r_array[i+1] != 0
        q = div(r_array[i], r_array[i+1])
        push!(r_array, r_array[i] % r_array[i+1])
        push!(x_array, x_array[i] - q * x_array[i+1])
        push!(y_array, y_array[i] - q * y_array[i+1])
        i += 1
    end

    return r_array[i]
end
```

Расширенный алгоритм Евклида

При вычисслении значений мы считаем, что у нас есть функция, которая является отображением, я записала её в виде отдельной фкнуии для удобства, само случайное отображение я брала из условия

```
function f(a, n)
    a = (a^2 + 5) % n
    return a
end
```

Функция f

Далее я прописала все парметры, которые используются для решения данной задачи

```
n = 1359331
c = 1
a = c
b = c
d = 1
```

Входные данные

После этого я приступила к написанию функции, реализующей p-алгортм Полланда, используя аглоритм описанный в теоретической части, важно заметить, что там присутствует опечатка и когда мы рассчитываем значение параметра b, то приммеляем отображение дважды. Также, так как значение b > a, то при нахождении НОД мы могли получиться отрицательное число, я добавила проверку на отрицательность и в таком случае поставила умножение d на -1

```
function pollard(d, n, a, b)
    while true
        if d > 1 && d < n
            return d
        elseif d == n
            return "Делитель не найден"
        elseif d == 1
            a = f(a, n)
            b = f(f(b, n), n)
            d = euclid(n, a-b)
            if d < 0
                d *= -1
            end
        end
    end
end
```

p-алгоритм Полланда

Закончив с написанием функций, я написала последние строки для запуска функции и записи результат её выполнения в переменную res, а после выод резальтата в терминал

```
res = pollard(d, n, a, b)
println(res)
```

Запуск функции и вывод результата

Запустив данную программу в терминале я получила следующий вывод, на последним выводом сверзу можно видеть ппредыдущий с проверочными println, которые я использовала для того, чтобы быстро понимать, где расчеты идут неверно

```
function pollard(d, n, a, b)
    while true
        if d > 1 && d < n
            return d
        elseif d == n
            return "Делитель не найден"
        elseif d == 1
            a = f(a, n)
            b = f(f(b, n), n)
            d = euclid(n, a-b)
            if d < 0
                d *= -1
            end
        end
    end
end
```

Результат в консоли

Выводы

В процессе выполнения работы, я реализовала разложение на множители для заданного числа, а именно реализовала p-алгоритм Полланда на языке программирования Julia.

Список литературы

::: Пособие по лабораторной работе 5 {file:///C:/Users/bermu/Downloads/lab06.pdf}